НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

«На правах рукопису»
УДК 004.891.2

«До захисту допущено»
Завідувач кафедри
_____ Тимощук О.Л.
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра
зі спеціальності 122 Компьютерні науки та інформаційні технології
на тему: «Система прогнозування курсу криптовалют на основі аналізу тональності новин»

Виконав:
студент ІІ курсу, групи КА-65м
Єрохін Гліб Вадимович _____

Науковий керівник:
доц.кафедри ММСА, к.т.н, доц.
Дідковська М.В. _____

Рецензент:
доц. кафедри програмного забезпечення
комп'ютерних систем ФПМ,
к.т.н., доц.
Заболотня Т.М. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ
2018

РЕФЕРАТ

Магістерська дисертація: 68 с., 9 рис., 24 таб, 2 додатки, 22 джерела.

Тема: «Система прогнозування курсу криптовалют на основі аналізу тональності новин»

Об’єкт дослідження – коливання курсу криптовалют

Предмет дослідження – методи виявлення тональності тексту, методи прогнозування курсів валют.

Мета роботи – створити систему прогнозу курсу криптовалют, її реалізувати та дослідити.

Метод дослідження – розгляд та аналіз методів прогнозу курсу криптовалют.

Актуальність – дослідження молодого і динамічного ринку криптовалют.

Результати роботи: – проведено аналіз основних методів прогнозу курсу криптовалют, створено та досліджено модифікацію, в якій у якості зовнішнього чинника використовується аналіз тональності.

Новизна роботи:

– розроблено спосіб прогнозу курсу криптовалют, у якому використовується аналіз тональності новин як зовнішній чинник, що впливає на курс.

Шляхи подальшого розвитку предмету дослідження – більш прогресивні методи аналізу часових рядів.

КРИПТОВАЛЮТИ, АНАЛІЗ ТОНАЛЬНОСТІ, ПРОГНОЗУВАННЯ КУРСУ, НОВИНИ, СОЦІАЛЬНІ МЕРЕЖІ

ABSTRACT

Topic: "System for prediction of cryptocurrencies rate based on sentiment analysis of news"

Graduate work: 68 pages., 9 Fig., 24 tab, 2 applications, 22 sources.

The object of the study is cryptocurrencies rate fluctuation.

Subject of research - models used in systems of forecasting the rate of cryptocurrencies, their types and modifications.

The purpose of the work is to create a system for forecasting the rate of cryptocurrencies, to implement it and to investigate it.

The method of research - review and analysis of methods for forecasting the rate of cryptocurrencies.

Relevance - research of the young and dynamic market of cryptocurrencies.

Results of work: - the analysis of the main methods of forecasting the rate of cryptacurrencies was performed; implemented and researched a modification, in which the sentiment analysis of news is used as an external factor.

Novelty of work:

- developed a method for forecasting the cryptocurrencies rate, which uses the sentiment analysis of news as an external factor which affects the rate;

Further development of the subject of research - more advanced methods of analysis of time series.

CRYPTOCURRENCIES, SENTIMENT ANALYSIS, RATE FORECAST, NEWS, SOCIAL NETWORKS

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

BTC – Bitcoin

VADER - Valence Aware Dictionary and sEntiment Reasoner

AR – Autoregressive

ARMA – Autoregressive Moving Average

ARMAX - Autoregressive Moving Average eXogenous

ВСТУП

Як відомо, криптовалюти є однією з найбільш популярних тем останнього часу. Велика кількість людей заробляє за допомогою криптовалют – деякі займаються так званим майнінгом, а деякі заробляють на коливанні їх курсу. Волатильність курсу криптовалют є дуже високою, тому актуальна інформація про можливі зміни курсу є вкрай необхідною. Найбільш актуальним джерелом інформації про криптовалюти є соціальні мережі, адже в них люди постійно їх обговорюють. Саме тому, кращею буде система прогнозування курсу, яка буде брати до уваги настрої людей щодо криптовалют.

Об’єктом дослідження в даній роботі є коливання курсу криптовалют

Предметом дослідження – методи виявлення тональності тексту, методи прогнозування курсів валют.

Отже, метою цієї роботи є створення системи прогнозування курсу криптовалют на основі тональності новин і записів в соціальних мережах, її реалізація та дослідження.

Виконано наступну роботу для досягнення:

- розглянуто існуючі алгоритми для прогнозування курсу криптовалют, виявлено їх переваги та недоліки;
- розглянуто існуючі алгоритми аналізу тональності тексту, вибрано найбільш оптимальний з них;
- розроблено архітектуру системи;
- реалізовано програмний продукт.

РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ В СФЕРІ КРИПТОВАЛЮТ

1.1 Основні поняття в сфері криптовалют

Криптовалюти - це нова концепція в глобальній економіці. Вони існують менше десяти років, і вони вже привернули велику увагу. Тим більше, що з 2013 року вони відчувають бурхливі зміни в обмінних курсах. Криптовалюти належать до групи віртуальних валют. Ми можемо розглядати криптовалюту як цифровий носій обміну, заснований на принципах криптографії, що дозволяє здійснювати безпечні, децентралізовані та розподілені економічні транзакції. Теоретичні основи криптокультур вже вперше викладені в Хаумом в 1983 році. Криптовалюти інтегрують електронні віртуальні гроші з принципами криптографії. Основним принципом криптовалюти є те, що жодна особа (або організація) не може прискорити або суттєво зловживати виробництвом певної валюти. Зазвичай лише вся визначена сума криптовалюти колективно виробляється цілою системою криптовалюти. Швидкість виробництва визначається вартістю, визначеною раніше, і є загальнодоступною. Криптовалютна система дозволяє практично безкоштовне перерахування криптовалютних одиниць (відомих як монети) між клієнтськими додатками через комп’терну peer-to-peer мережу. Найбільш відома криптовалюта і перша, що була введена - Bitcoin у 2009 році. Його спроектовано людиною або групою людей, що ховаються під псевдонімом Сатоші Накамото. Існують два типи користувачів Bitcoin: звичайні користувачі та так звані майнери. Звичайні користувачі Bitcoin використовують цифровий гаманець, подібний до електронних банківських додатків. Гаманець - це програмне забезпечення для управління Bitcoin, отже, для відправлення та отримання платежів в Bitcoin. Bitcoin існують лише як інформація в файлах на комп'ютері або на мобільному пристрої. Доступ до цих файлів обмежується власником приватного ключа, який використовується для безпеки грошей. Якщо файлова система на комп'ютері

пошкоджена або файл випадково видалений, тоді файл гаманця втрачається, а Bitcoin, що містяться в ній втрачаються назавжди (у випадку, якщо не було створено резервного файлу). Якщо не буде зламано дуже складний алгоритм шифрування, вбудоване в систему, то неможливо буде відновити втрачені монети, а зламати шифрування, яке використовує мережа Bitcoin силою, практично неможливо за достатню кількість часу.[1]

Однак, в останній час (починаючи з 2013 року) з'явилась ще одна область використання криптовалют. У 2009 році, коли криптовалюта тільки з'явилась, ніхто про неї не знав і не користувався. Тому її вартість була рівна $0. Навіть за наступний рік вартість Bitcoin не перевищувала $0.39. Однак, протягом 2013 року вартість 1 Bitcoin перевищила $1000, а протягом 2017 його вартість іноді була близько $20000. Очевидно, що для багатьох людей це стало дещо схожим на ринок акцій, люди почали купувати криптовалюти для того, щоб потім на цьому заробити. Однак, в кінці 2017 року вартість знизилась до майже $13000, що означає, що волатилність є дуже високою. Саме тому задача прогнозування курсу криптовалют є дуже актуальною.

На те, що у курсу настільки висока волатильність є декілька причин:

- Перший фактор полягає в тому, що криптовалюти мають менші розміри ринку у порівнянні з встановленими формами валюти. Це означає, що навіть невеликі рухи криптовалют можуть мати видимий вплив на їх ціну. Додаючи до цього, розподіл багатства у сфері криптовалют ще більш перекручений, ніж традиційне багатство, тому люди з великими частками в криптовалюті мають непропорційно велику владу над своїми цінами.
- Другий фактор полягає в тому, що громадське сприйняття криптовалют (зокрема, Bitcoin) є досить дихотомічним. Наприклад, подорожчання у другому кварталі 2017 року може бути пов'язане з низкою факторів (наприклад, збільшення інтересу в усіх країнах Азії, збільшення прийняття підприємств, Litecoin успішно активізує SegWit, щоб збільшити пропускну спроможність транзакції), які збільшили позитивні настрої у цьому

просторі. З іншого боку, негативні зміни в законності криптовалют, а також злами системи можуть дуже легко знищити їхні суспільні інтереси.[2]

Через те, що криптовалюти мають високу волатильність, задача прогнозування їх курсу є дуже важливою.

Далі приведені основні методи прогнозування курсу.

1.2 Алгоритми прогнозування курсу криптовалют

1.2.1 Алгоритм на основі пошуку K найближчих сусідів

Алгоритм був реалізований у вигляді Twitter-бота, який кожні 2 години буде постити прогнози щодо значення курсу Bitcoin протягом наступних N днів. N - це кількість днів, коли люди запитують найбільше. Так, наприклад, якщо 3 людини запитували, щоб бот прогнозував значення протягом наступних 5 днів, а 7 людей просили, щоб бот прогнозував значення впродовж наступних 2 днів, бот постив прогноз на 2 дні, тому що більше людей просили це передбачення.

Цей алгоритм прогнозування побудований з наступних кроків:

1. Збір усіх запитів.
2. Отримання найбільш запитуваної кількості днів, для передбачення.
3. Отримання поточного значення курсу біткойна.
4. Знаходження K найближчих дат за останні 2 місяці, коли значення біткойну було найбільш схожим на поточне значення.
5. Пошук значень BTC для кожної знайденої дати (назвемо PAST_DATE) після наступних N днів (назвемо N_DAYS_AFTER_PAST_DATE).
6. Обчислення різниці між значеннями N_DAYS_AFTER_PAST_DATE і PAST_DATE для кожної дати.
7. Обчислення середнього значення для різниць.

8. Отриманий результат показує скільки биткойн в середньому виріс за заданий проміжок часу між усіма періодами PAST_DATES і N_DAYS_AFTER_PAST_DATES.

За словами автора алгоитму, прогноз не завжди відповідає правильному значенню, але в більшості випадків похибка складає близько 100-200 $.

Проблема з цим алгоритмом полягає в тому, що він просто аналізує історичні дані біткойна та робить прогноз орієнтуючись на них. [3]

1.2.2 Використання Байесівської регресії

Алгоритм, запропонований Devavrat Shah та Kang Zhang використовує Байесівську регресію з моделлю прихованого джерела для передбачення курсу bitcoin. Автори використовують дані з інтервалом у 10 секунд для того, щоб підказати користувачу оптимальну стратегію. Торгова стратегія дуже проста: у кожний момент підтримується позиція з +1 bitcoin, 0 bitcoin або -1 bitcoin. В кожний момент часу, прогнозується середній рух цін за наступні 10 секунд (скажімо, Δp), використовуючи байєсівську регресію, і якщо Δp> t (поріг), то необхідно купувати bitcoin, якщо поточна позиція ≤ 0; якщо Δp <-t, то необхідно продавати bitcoin якщо поточна позиція ≥ 0; в іншому випадку нічого не робити. Вибір часових кроків, в які приймаєть торговельні рішення згадані вище, вибираються ретельно, дивлячись на останні тенденції.

Основний метод прогнозу середньої зміни ціни Δp за 10-секундний інтервал - це байєсівська регресія. Враховуючи, що часовий ряд зміни вартості bitcoin протягом інтервалу у декілька місяців, вимірюється кожні 10 секундн, ми маємо дуже великий часовий ряд (або вектор). Використовуючи цей історичний часовий ряд, звідси генеруються три підмножини даних часового ряду з трьома

різними довжинами: S_1 - за 30 хвилин, S_2 - за 60 хвилин та S_3 - за 120 хвилин. Тепер у певний момент часу, щоб прогнозувати майбутню зміну Δp, використовуються історичні дані трьох довжини: попередні 30 хвилин, 60 хвилин та 120 хвилин - позначаються x_1, x_2 та x_3. Ми використовуємо x_j з історичними зразками S_j для Байєсівської регресії для прогнозування середньої зміни ціни Δp_j для 1 ≤ j ≤ 3. Ми також обчислимо $r = (v_{bid} - v_{ask}) / (v_{bid} + v_{ask})$, де v_{bid}є загальний обсяг, який люди готові купити у найвищих 60 замовлень і v_{ask}- це загальний обсяг, який люди готові продати у 60 найвищих замовленнях на основі поточних даних замовлень (отримані з Okcoin.com). Остаточна оцінка Δp обчислюється як $\Delta p = w_0 + \sum_{j=1}^{3} w_j \Delta p_j + + w_4 r$, де $w = (w_0, \ldots, w_4)$ - вивчені параметри.

Для пошуку S та w весь проміжок часу розбивається на три приблизно однакові періоди. Перший період використовується щоб знайти шаблони S_j, 1 ≤ j ≤ 3. Другий період використовується для вивчення параметрів w, а останній третій період використовується для оцінки ефективності алгоритму. Вивчення w виконується просто шляхом пошуку найкращої лінійної функції над усіма S_j, 1 ≤ j ≤ 3. Для вибору S_j, 1 ≤ j ≤ 3 беруться всі можливі часові ряди відповідних довжин (вектори розмірності 180, 360 і 720 відповідно для S_1, S_2 і S_{j3}). Кожна з них формує x_i та відповідну мітку y_i обчислюється шляхом перегляду середньої зміни ціни в 10-секундному інтервалі після закінчення x_i. Шаблони були згруповані в 100 кластерів, використовуючи алгоритм k-means. З них були обрані 20 найбільш ефективних кластерів і були взяті репрезентативні шаблони з цих кластерів.[4]

На рис. 1.1 показаний результат роботи алгоритму, в якому синя лінія – курс bitcoin, зелені точки – точки, в які алгоритм пропонував продати bitcoin, червоні точки – коли пропонував купити. [5]

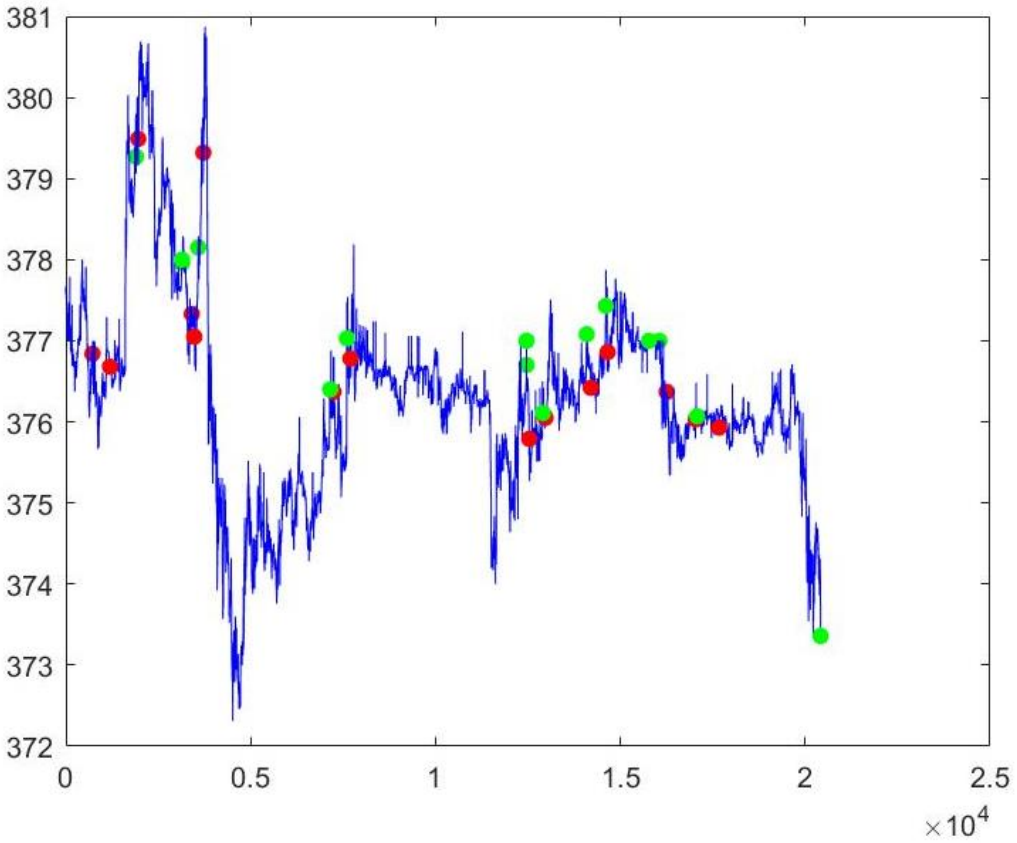


Рисунок 1.1 – Результат роботи алгоритму[5]

1.3 Прийняття до уваги новин та записів в соціальних мережах

Існує думка, що більшість людей, які купують/продають криптовалюти в першу чергу орієнтуються не на числове значення курсу, а на реакцію і популярність об’єкту, у який вони вкладають гроші у соціальних мережах. Можна сказати, що більшість цих людей мають доступ до інтернету, є активними користувачами соціальних медіа, читають форуми, блоги та сайти, а також рідко дивляться телебачення. Тож їх рішення мають спиратися на те, що відбувається в інтернеті. Якщо відобразити на графіку курс bitcoin, а також кількість його згадувань у соціальних медіа отримаємо графік, показаний на рис 1.2.

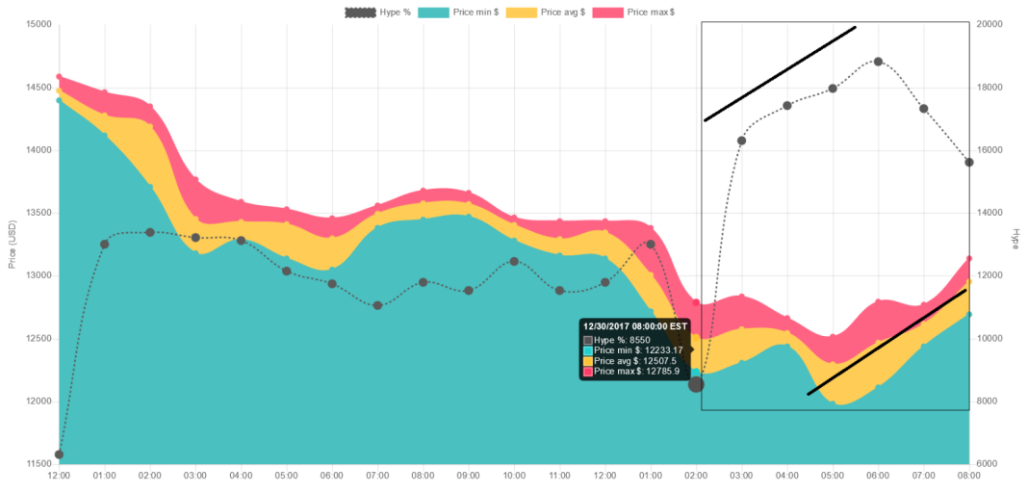


Рисунок 1.2 – Згадування про BTC та його курс

Слід зазначити, що дані графіки є відносними, однак, можна побачити, що між ними є деяка залежність, наприклад, зріст згадувань при різкому падінні курсу о 01.00. [6]

Однак, окрім аналізу кількості записів має значення тональність тексту – наскільки позитивна чи негативна думка автора. Для аналізу цього була введена метрика, що рахує різницю між позитивними та негативними словами у тексті. Список позитивних і негативних слів був взятий зі спеціального маркованого списку, створеного для аналізу даних ринків акцій.

Результат роботи показаний на рис. 1.3.

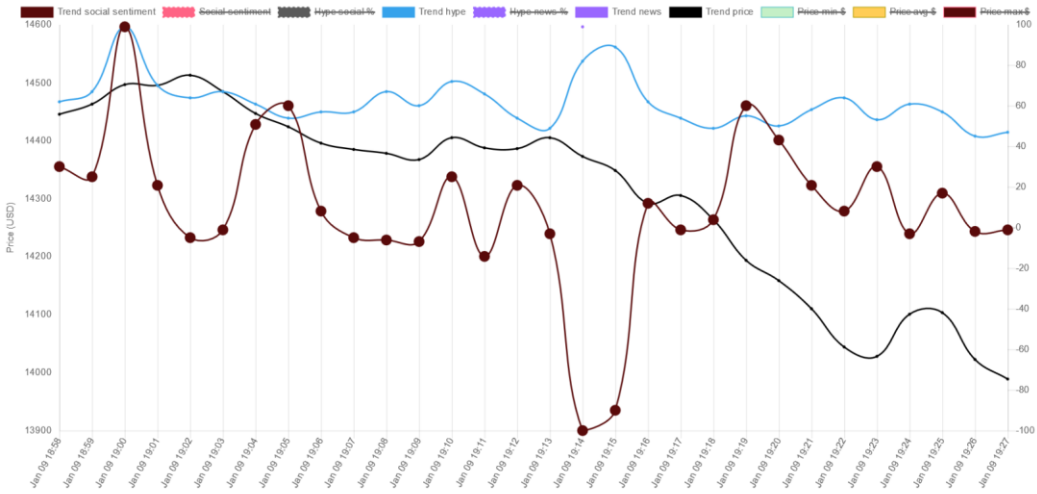


Рисунок 1.3 - Графік курсу BTC та тональності тексту

Блакитним кольором позначена кількість згадувань, коричневим – тональність, а чорним – курс BTC.[7]

Висновки до розділу

В розділі розглянуто основні поняття в сфері криптовалют, що стосуються даної роботи. Описано саме поняття криптовалют, та основні алгоритми, що використовуються для прогнозування курсу. Основним недоліком цих алгоритмів є те, що вони використовують тількі історичні дані про курс, а не ситуацію в даний момент. Розглянуто підхід, який враховує настрій в соціальних мережах як метод прогнозування курсу.

РОЗДІЛ 2. ОСНОВНІ ПОНЯТТЯ В СФЕРІ АНАЛІЗУ ТОНАЛЬНОСТІ ТЕКСТУ

У 2017 році люди світу генерують 2,5 мільйона терабайт інформації на день [8]. 500 мільйонів твітів, 1,8 мільярда записів на Facebook, кожен день [9]. Ці шматки інформації описують все, що відбувається у світі; від того, хто що їв обід, до їхньої ненависті до рефері у футбольному матчі. Twitter став відомим як місце, де новини швидко розповсюджуються в стислій формі. Що стосується фінансового ринку, то суспільна довіра до певного товару є основною базою її вартості. Соціальні медіа служили платформою для висловлення думки з моменту їх створення, і таким чином, використовуючи відкриті API, подібні до Facebook і Twitter, ці очевидно необ'єктивні фрагменти інформації стають доступними з морем метаданих.

У статті «Трейдинг у Twitter: використання тональності соціальних медіа для прогнозування повернення акцій»[10] 2,5 мільйона твітів про фірми, що входять в S & P 500 були класифіковані авторами власного класифікатора тональності і порівнювалися з курсом акцій. Результати показали, що тональність, яка розповсюджується через соціальну мережу, як очікується, буде відображатись на ціні акцій у той же торговий день, тоді як більш повільне поширення настроїв буде відображатися в майбутніх торговельних днях. Якщо базувати стратегію трейдингу, на цих прогнозах, передбачається досягтнення 11-15% прибутків на рік. У статті «Алгоритмічна торгівля криптовалютою на основі аналізу тональності Twitter» [11] аналогічно проаналізовано, яким чином тональність твітів може буди використана для впливу на інвестиційні рішення, зокрема для Bitcoin. Автори використовували технології машинного навчання з учителем, що дало кінцеву точність вище 90% погодинно і по днях. Автори зазначають, що точність 90% була отримана шляхом надійного аналізу помилок на вхідних даних, що в середньому збільшило точність на 25% кращу точність.

Коліанні разом із Хатто та Гілбертом згадували про рівні шуму в своєму наборі даних, а попередня команда отримала значне зниження рівня помилок після очищення свого набору даних від шумів.

Аналіз тональності є задачею обробки природної мови. Обробка природних мов пов'язана з областью взаємодії комп'ютера людини. Завдання ідентифікації думки огляду називається аналізом настроїв. Думка може бути позитивною, негативною чи нейтральною.[12]

Метою аналізу тональності є знаходження думок в тексті і визначення їх властивостей. Залежно від поставленого завдання нас можуть цікавити різні властивості, наприклад:

- автор - кому належить ця думка
- тема - про що йдеться на думці
- тональність - позиція автора щодо згаданої теми (зазвичай «позитивна» або «негативна»)

У літературі зустрічаються різні способи формалізувати модель думок. Також використовується і різна термінологія. В англійській мові цю область дослідження зазвичай називають opinion mining and sentiment analysis (дослівно: «пошук думок і аналіз почуттів»). В україномовних джерелах зазвичай вживається термін «аналіз тональності». Незважаючи на те, що тональність є лише однією з характеристик думки, саме завдання класифікації тональності є найбільш часто досліджуваним в наші дні. Це можна пояснити кількома причинами:

- Визначення автора і теми є набагато більш важкими завданнями ніж класифікація тональності, тому має сенс спочатку вирішити більш просту задачу, а потім вже переключитися на інші.
- У багатьох випадках нам досить лише визначити тональність, тому що інші характеристики нам уже відомі. Наприклад, якщо ми збираємо думки з блогів, зазвичай авторами думок є автори постів, тобто визначати автора нам не потрібно. Також часто нам вже відома тема: наприклад, якщо ми

виробляємо в Твіттері пошук за ключовим словом «Windows 8», то потім нам потрібно лише визначити тональність знайдених твітів. Звичайно ж, це працює не у всіх випадках, а лише в більшості з них. Але ці припущення дозволяють в значній мірі спростити і так нелегке завдання.[13]

Основні кроки аналізу тональності представлені на рис. 2.1.[12]

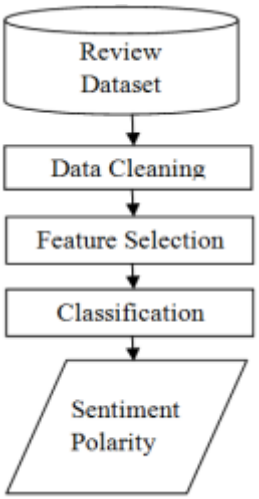


Рисунок 2.1 – Основні кроки аналізу тональності

2.1 Попередня обробка даних

Наступні методи використовуються в аналізі тональності як частини попередньої обробки.

- Перетворення верхніх та нижніх літер, вилучення небжаної пунктуації, видалення розривів рядків, видалення спеціальних символів, видалення коду ASCII, видалення зайвих пробілів тощо.
- Стеммінг – алгоритм визначення основи слова. Стеммер Портера-найпоширеніший алгоритм для стеммінгу.
- Правило відхилення: цей метод видаляє заперечення слова, яке змінює значення слова на протилежне.

– Правило кон'юнктури: цей метод витягує сенс за допомогою граматичного правила.[12]

2.2 Вибір ознак

Найбільш поширений спосіб представлення документа в задачах комп’ютерної лінгвістики і пошуку - це або у вигляді набору слів (bag-of-words) або у вигляді набору N-грам.[13]

Підхід bag-of-words можна представити наступним чином. Кожному слову, що представлене в наборі документів записати у вектор f. Тоді кожен документ можна представити у вигляді $d = (n_1(d), n_2(d), \dots, n_m(d))$, де d – документ, m – кількість слів (ознак), $n_j(d)$ – кількість входжень слова f_j у документ d.[12]

Підхід з набором N-грам є схожим на попередньо описаний, окрім того, що у вектор f входять послідовності слів довжиною N.

Інший спосіб представлення тексту - символьні N-грами. Незважаючи на те, що такий спосіб може здатися занадто примітивним, тому що на перший погляд набір символів не несе в собі ніякої семантики, проте цей метод іноді дає результати навіть краще ніж N-грами слів. Якщо придивитися, то можна побачити, що N-грами символів відповідають в якійсь мірі морфемам слів, а зокрема корінь слова несе в собі його зміст. Символьні N-грами можуть бути корисні в двох випадках:

– при наявності орфографічних помилок в тексті - набір символів у тексту з помилками і набір символів у тексту без помилок буде практично однаковий на відміну від слів;
– для мов з багатою морфологією (наприклад, для російської, української) - в текстах можуть зустрічатися однакові слова, але в різних варіаціях (різні

рід або число), але при цьому корінь слів є незмінним, а отже і загальний набір символів також.

Символьні N-грами застосовуються набагато рідше ніж N-грами слів, але іноді вони можуть поліпшити результати.

Також можна використовувати додаткові ознаки, такі як: частини мови, пунктуація (наявність в тексті смайлів, знаків оклику), наявність в тексті заперечень («не», «ні», «ніколи»), вигуків і т.д.[13]

Для остаточного складання вектору ознак необхідно додати вагу до кожної ознаки. Найпростіший підхід показаний вище, у ньому кожній ознаці ставиться у відповідність кількість входжень її у документ. Більш цікавим способом є TF-IDF. У ньому кожній ознаці ставиться у відповідність її вага, що розраховується наступним чином (формула 2.1):

$$V_{t,d} = C_{t,d} * \log(\frac{|N| * P_t}{|P| * N_t}) \quad (2.1)$$

де:

- $V_{t,d}$ – вага ознаки t в документі d
- $C_{t,d}$ – кількість входжень ознаки t в документ d
- $|P|$ – кількість документів з позитивною тональністю
- $|N|$ – кількість документів з негативною тональністю
- P_t – кількість позитивних документів, в яких зустрічається ознака t
- N_t– кількість негативних документів, в яких зустрічається ознака t

В результаті вага слів з позитивною тональністю буде великим позитивним числом, вага слів з негативною тональністю буде негативним числом, вага нейтральних слів буде близький до нуля. Таке зважування вектора ознак в більшості випадків дозволяє поліпшити точність класифікації тональності.[13]

2.3 Методи машинного навчання

Найбільш використовуваними класифікаторами для аналізу тональності є SVM (метод опорних веторів) та наївний байесівський класифікатор.

2.3.1 SVM

Метод опорних веторів вивчає дані, визначає гіперплощину, яка класифікує дані в два класи з максимальним запасом. SVM також підтримує класифікацію та регресію в статистичному навчанні. Відокремлююча гіперплощина записується так (формула 2.2):

$$W * X + b = 0 \tag{2.2}$$

де:

- $W = \{w_1,\ w_2, \dots, w_n\}$ – вектор ваг для n атрибутів
- b – білий шум

Відстань від відокремлюючої гіперплощини до будь-якої точки на H1 становить 1 / | W | і так само до будь-якої точку на H2 становить 1 / | W |. Таким чином максимальний запас 2 / | W |. Якщо значення гіперплощини > 0 то класифікуємо як позитивну категорію, якщо значення гіперплощини <0 то як негативну, якщо значення гіперплощини = 0, то всі точки перпендикулярні до W. Якщо значення запасу велике, великі штрафи присвоюються помилкам / помилкам запасу. Якщо значення запасу невелике, то деякі пункти стають похибкою, а орієнтація гіперплощини змінюється, як показано на формулі 2.3

$$W = \sum_j \alpha_j c_j d_j , \alpha_j \geq 0 \quad (2.3)$$

де $c_j \in \{-1,1\}$ – клас документа d_j.[12]

2.3.2 Наївний байесівський класифікатор

Він використовується для прогнозування ймовірності того, що певний елемент належить до певного класу. Він використовується через його легкість як під час тренувань, так і під час класифікації. Заздалегідь оброблені дані подаються як вхід для навчання, наївного байесівського класифікатора, і модель застосовується до тестових даних для отримання позитивної чи негативної тональності. Теорема Байеса полягає в наступному (формула 2.4).

$$P(H|X) = \frac{P(X|H)*P(H)}{P(X)} \quad (2.4)$$

де H – гіпотеза, X – елемент вибірки.[12]

2.4 Підходи, що засновані на словниках

Значна частина підходів до аналізу тональності залежить від лексикону, що лежить в основі тональності (або думки). Лексикон тональності - це список лексичних ознак (наприклад, слів), які зазвичай позначаються відповідно до їх семантичної орієнтації як позитивні або негативні (Лю, 2010). Ручне створення

та перевірка таких списків основних рис, є одночасно одним з найбільш надійних методів для створення надійних лексиконів тональності, але й одним з найбільш трудомістких. З цієї причини значна частина прикладних досліджень в сфері аналізу тональності, спирається на вже існуючі словники, створені вручну. Оскільки словники є дуже корисними для аналізу тональності, далі йде огляд кількох з них. Спочатку розглядаються три лексикони, що широко використовуються (LIWC, GI, Hu-Liu04), в яких слова класифікуються у бінарні класи (тобто позитивні або негативні) відповідно до їх контекстно-вільної семантичної орієнтації. Потім описуються три інші лексики (ANEW, SentiWordNet та SenticNet), в яких слова пов'язані з валентними оцінками інтенсивності тональності.

2.4.1 Бінарні словники

LIWC - це програмне забезпечення для аналізу тексту, призначеного для вивчення різних текстових зразків емоційних, когнітивних, структурних та технологічних компонентів. LIWC використовує власний словник з майже 4500 слів, організованих в одну (або більше) з 76 категорій, включаючи 905 слів у двох категоріях, особливо пов'язаних з аналізом тональності (табл. 2.1):

Таблиця 2.1 - Категорії слів LIWC

Категорія LIWC	Приклади	К-сть слів
Позитивна емоція	Love, nice, good, great	406
Негативна емоція	Hurt, ugly, sad, bad, worse	499

LIWC добре зарекомендував себе і пройшов внутрішню та зовнішню перевірку у процесі, що триває більше десяти років, включаючи роботу психологів, соціологів та лінгвістів (Pennebaker et al., 2001; Pennebaker et al., 2007). Той факт, що цей словник перевірений роками, робить LIWC привабливим варіантом для дослідників, які шукають надійний лексикону, щоб витягнути почуття, емоційність або настрій з тексту соціальних мереж. Наприклад, лексикон LIWC був використаний для виявлення показів політичних настроїв від твітів (Tumasjan, Sprenger, Sandner, & Welpe, 2010), передбачення початку депресії у людей на основі тексту соціальних медіа (De Choudhury, Gamon, Counts, & Horvitz, 2013), характеристики емоційної мінливості вагітних з постів Twitter (De Choudhury, Counts, & Horvitz, 2013), ненав'язливого вимірювання національного щастя на основі оновлень статусу Facebook (Kramer, 2010) та розрізнення щасливих романтичних пар від нещасливих на основі їх миттєвих повідомлень (Hancock, Landrigan, & Silver, 2007). Проте, як зазначають Hutto, Yardi та Gilbert (2013), незважаючи на широке використання його для оцінки тональності у тексті соціальних мереж, LIWC не включає в себе розгляд таких лексичних елементів, як акронім, ініціалізм, смайлики або сленг, які, як відомо, важливі для аналізу тональності текстів у соціальних мережах (Davidov, Tsur, & Rappoport, 2010). Також, LIWC не в змозі брати до уваги інтенсивність тональності слів. Наприклад, "тут виняткова їжа", має більше позитивну інтенсивність, ніж "тут нормальна їжа". Інструмент аналізу тональності, який використовує LIWC, оцінить їх однаково (кожна з них містить один позитивний термін).

Загальний запит (General Inquirer, GI) - це програма для аналізу тексту з одним із найстаріших вручну побудованих лексиконів, які все ще широко використовуються. GI розвивається і вдосконалюється з 1966 року і розробляється як інструмент контент-аналізу, який використовується соціологами, політологами та психологами для об'єктивної ідентифікації зазначених характеристик повідомлень (Stone та ін., 1966). Лексикон містить

більше 11000 слів, які класифікуються в одну або більше з 183 категорій. Для наших цілей ми зосереджуємося на 1915 слів, позначених як «позитивне», і 2291 слів, позначених як «негативне». Як і LIWC, Гарвардський словник GI широко використовувався в декількох роботах, щоб автоматично визначати властивості тональності тексту (Esuli & Sebastiani, 2005; Kamps, Mokken, Marx & de Rijke, 2004; Turney & Littman, 2003). Однак, як і у випадку з LIWC, GI страждає від недостатнього охоплення релевантними для сприйняття лексичними властивостями, що є загальними для текстів у соціальних мережах, і не знає різниці в інтенсивності між словами, які мають тональність.

Hu та Liu (Hu & Liu, 2004; Liu, Hu, & Cheng, 2005) підтримують загальнодоступну лексику, що скалдається з майже 6800 слів (2000 з позитивною семантичною орієнтацією та 4783 з негативнимою). Їх лексикон спочатку був побудований за допомогою процесу бутстрепінгу (Hu & Liu, 2004) з використанням WordNet (Fellbaum, 1998), відомої англійської лексичної бази даних, в якій слова сгруповані в групи синонімів, відомі як синсети (синонимічні множини). У минулому десятилітті словник Hu-Liu04 розвивався, і (на відміну від LIWC або GI-лексиконів) більше підходить для аналізу тональності у соціальному тексті та оглядах продуктів, хоча він все ще не фіксує почуття від смайлів або абревіатур.

2.4.2 Валентні словники

Багато додатків отримають користь, якщо будуть мати змогу визначити не тільки бінарну полярність (позитивний або негативний), а також силу почуттів, виражених у тексті. Про те, наскільки сприймають чи не спрймають люди новий продукт, фільм чи закон? Аналітики та дослідники прагнуть (і повинні) знайти зміни інтенсивності настроїв з плином часу, щоб виявити, коли риторика є

сильнішою або слабшою (Wilson, Wiebe, & Hwa, 2004). Корисним буде мати загальну лексику з силовими валентностями.

Лексикон «Афективні норми для англійських слів» (Affective Norms for English Words, ANEW) має набір нормативних емоційних рейтингів для 1034 англійських слів (Bradley & Lang, 1999). На відміну від LIWC або GI, слова ANEW оцінюються за рівнем задоволення, збудження та домінантності. Слова ANEW мають зв'язану валентність настрою від 1 до 9 (з нейтральною точкою зі значенням у п'ять), такими, що слова з оцінками валентності менше п'яти вважаються неприємними / негативними, а ті, що мають оцінку більшу, ніж п'ять, розглядаються як приємні / позитивниі. Наприклад, валентність для слова зрадництво (betray) складає 1,68, м'яка (bland) - 4,01, мрія (dream) - 6,73, а захоплення (delight) - 8,26. Ці валентності допомагають дослідникам вимірювати інтенсивність виражених настроїв у мікроблогах (De Choudhury, Counts та ін., 2013; De Choudhury, Gamon та ін., 2013; Nielsen, 2011) - важливий вимір за рамками простої бінарної орієнтиції «позитивний чи негативний». Тим не менше, як і у випадку з LIWC та GI, лексикон ANEW також нечутливий до загальних лексичних особливостей соціального тексту.

SentiWordNet - це розширення WordNet (Fellbaum, 1998), в якому 147 306 синонімів анотовані з трьома числовими оцінками, що стосуються позитивності, негативності та об'єктивності (нейтральності) (Baccianella, Esuli, та Sebastiani, 2010). Кожен бал знаходиться в діапазоні від 0,0 до 1,0, а їх сума становить 1,0 для кожної синонімічної множини. Оцінки були розраховані, використовуючи складну суміш алгоритмів з учителем і без (методи пропагації та класифікатори). Таким чином, це не є джерелом золотого стандарту, таким як WordNet, LIWC, GI або ANEW (котрі на 100% були створені людьми), але це є корисним для широкого кола завдань. Лексикон SentiWordNet є дуже зашумленим; значна частина синонімічних множин не має позитивної чи негативної полярності. Також не враховуються лексичні особливості, що стосуються тексту в мікроблогах.

SenticNet - це загальнодоступний семантичний та афективний ресурс для аналізу тональності на рівні концепцій (Cambria, Havasi, & Hussain, 2012). SenticNet побудований за допомогою парадигми Sentic Computing (від sentence - речення), яка використовує технології штучного інтелекту та Semantic Web для обробки природних мов використовуючи ансамбль алгоритмів аналізу графів та методів зменшення розмірності (Cambria, Speer, Havasi, & Hussain, 2010) . Лексикон SenticNet складається з 14244 концепцій здорового глузду, таких як гнів, обожнювання, горе та захоплення, з інформацією, пов'язаною, серед іншого, з полярністю тональності концепту, числове значення в безперервному діапазоні від -1 до 1.

2.4.3 Словники та контекст слова

Незалежно від того, використовують словники бінарну полярність або більш точні валентні лексикони, можна покращити ефективність аналізу тональності, розуміючи більш глибокі лексичні властивості (наприклад, частини мови) для більшого розуміння контексту. Наприклад, лексикон може бути додатково підлаштований згідно з процесом розбіжностей у словах (Word-sense disambiguation, WSD) (Akkaya, Wiebe, & Mihalcea, 2009). WSD означає процес визначення того, яке зі значень слова використовується у реченні, коли слово має декілька значень (тобто його контекстне значення). Незважаючи на розповсюдженість підходів до аналізу настроїв на основі лексики у контексті соціальних мереж, вони мають два основні недоліки:

1. Проблеми з охопленням, алгоритми часто ігнорують важливі лексичні особливості, які особливо важливі для соціального тексту в мікроблогах
2. Деякі словники ігнорують загальні настрої інтенсивності для рис в межах лексикону;

Створення нового комплекту (золотого стандарту, підтвердженого людьми) лексичних ознак, разом з відповідними оцінками валентності настрою, може бути дуже трудомістким процесом.

2.4.4 Алгоритм VADER

Алгоритм VADER – валентний алгоритм, що був створений спеціально для аналізу записів в соціальних мережах. Підхід спрямований на використання переваг незалежного моделювання, основаного на правилах, для побудови двигуна для обчислення тональності, який

а) Добре працює в середовищі соціальних мереж, але легко узагальнюється на інші домени;

б) Не вимагає даних для навчання, але побудований з загальноприйнятим, заснованим на валентній основі, створеним людиною лексиконі;

в) Достатньо швидкий для використання в режимі он-лайн з потоковими даними;

г) Не сильно страждає від компромісу швидкісного виконання.

2.4.4.1 Створення лексикону

Створення (а тим більше - валідація) повного лксикону для аналізу тональності - це трудомісткий процес, в якому часом можуть зустрічатися помилки, тому не дивно, що багато дослідників та практиків, які займаються дослідженням тональності, настільки сильно покладаються на наявні лексикони.

Існує, звичайно, велика частка дублікатів у словниках, що охоплюється такими лексиконами; однак, є також численні входження, унікальні для кожного. Автори алгоритму почнали з побудови списку слів, взятого з існуючих і добре зарекомендувавших себе словників (LIWC, ANEW, GI). До цього додали численні лексичні входження, що є спільними для вираження настроїв у соціальних мережах, включаючи повний список смайлів (наприклад, ":-)" означає "посмішку" і в цілому вказує на позитивні настрої), абревіатури, що пов’язані з настроями (наприклад, LOL і WTF мають явно виражену тональність) та широко використовуваний сленг, який має тональність (наприклад, "nah", "meh" та "giggly"). Цей процес надав розробникам більше 9000 кандидатів для дослідження.

Далі розробники оцінили загальну придатність кожного кандидата до виражень настроїв. Використовувався колективний підхід (wisdom-of-the-crowd) (WotC), щоб отримати дійсну точну оцінку валентності (інтенсивності) настроїв кожного контекстного кандидата. Було зібрано рейтинги інтенсивності для кожної з лексичних характеристик з десяти незалежних оцінювачів (в загальному більше 90000 оцінок). Характеристики оцінювалися за шкалою від «[-4] Надзвичайно негативний» до «[4] надзвичайно позитивний», з урахуванням" [0] нейтральний". Було збережено кожну лексичну характеристику, яка мала середній рейтинг, відмінний від нуля, і стандартне відхилення якого становило менше 2.5, визначене сукупністю десяти незалежних оцінок. Цей процес залишило трохи більше 7500 лексичних функцій з перевіреними значеннями валентності, які вказують на полярність почуття (позитивний / негативний) та інтенсивність настроїв у масштабі від -4 до +4. Наприклад, слово «гаразд» (“okay”) має позитивну валентність 0,9, «добре» (“good”) має 1,9, «чудово» (“great”) - 3,1, «жахливо» (“horrible”) - 2,5, сумний смайлик «:(» - це -2,2, а «смокче» (“sucks”, “sux”) - це -1,5.

Попередні лінгвістичні експерименти із оцінкою, які використовують підхід WotC до створення словників, виявилися надійними - іноді вони дають

кращі результати за експертів. З іншого боку, попередня робота також дала поради щодо методів зменшення шуму від людей, які беруть участь в оцінюванні, які можуть давати неправильні результати. Розробники запровадили чотири процеси контролю якості, щоб забезпечити отримання важливих даних від оцінювачів.

По-перше, кожен читач був попередньо перевірений на рівен розуміння англійської мови - кожен оцінювач мав отримати оцінку у 80% або вище у стандартному тестуванні на читання на рівні коледжу.

По-друге, кожному оцінювачу довелося завершити тренінг по онлайн-оцінюванню тональності та отримати оцінку у 90% або вище у тесті що включав в себе окремі слова, смайли, акроніми, речення, твіти, та фрагменти тексту (наприклад, фрагменти речення або фрази).

По-третє, кожна порція з 25 входжень містила п'ять входжень з відомим (попередньо підтвердженим) рейтингом тональності. Якщо оцінювач відхилився більше ніж на значення одного стандартного відзилення від середнього значення цього відомого рейтингу на трьох або більше з п'яти входжень, розробники відмовлялися від всіх 25 оцінок, отриманих від цього оцінювача.

Нарешті, розробники запровадили бонусну програму для стимулювання та винагороди високої якості роботи. Наприклад, оцінювачів попросили вибрати баланс валентності, який, на їх думку, "більшість інших людей" вибере за даною лексичною ознакою. Працівникам компенсували $0,25 за кожну партію з 25 предметів, які вони оцінили, з додатковим бонусом за стимулом від $0,25 для всіх працівників, які успішно підібрали середнє значення групи (у межах 1,5 стандартних відхилень) принаймні 20 з 25 відповідей у кожній партії.

Використовуючи ці чотири методи контролю якості, було досягнуто високої цінності даних, отриманих від оцінювачів - стимулюючі бонуси за високу якість отримали близько принаймні 90% оцінювачів.

2.4.4.2 Додаткові показники

Було проаналізовано спаціальну вибірку із 400 позитивних та 400 негативних текстових фрагментів з соціальних мереж (твітів). Ці зразки було обрано з набору 10-ти тисяч випадкових твітів, тональність яких була оцінена, використовуючи двигун аналізу настроїв Pattern.en (вибрано 400 найбільш позитивних і 400 найбільш негативних записів). Pattern.en - набір інструментів для обробки природної мови (NLP), який використовує WordNet для оцінки думки відповідно до англійських прикметників, що використовуються в тексті.

Далі, два експерти в індивідуальному порядку розглянули всі 800 твітів і самостійно оцінили інтенсивність настроїв у масштабі від -4 до +4. Слідом за методом індукційного кодування, керованим даними, подібним до підходу "Заземлені теорії", було використано якісний аналіз методів ідентифікації властивостей і характеристик тексту, які впливають на інтенсивність настроїв тексту. Цей глибокий якісний аналіз призвів до ізоляції п'яти загальмовуваних евристик на основі граматичних та синтаксичних сигналів, щоб передати зміну інтенсивності настроїв. Важливо, що ця евристика виходить за рамки того, що, як правило, є зафіксовано в типовому наборі слів. Вони містять приховані відносини між словами:

1. Пунктуація, а саме знак оклику (!), збільшує величину інтенсивності, не змінюючи семантичної орієнтації. Наприклад, фраза "їжа тут добра!!!" має більшу інтенсивність, ніж "їжа тут добра".

2. Використання верхнього регістру слів, щоб підкреслити відповідне слово, за наявності інших слів, збільшує інтенсивність настроїв, не впливаючи на семантичну орієнтацію. Наприклад, "їжа тут ЧУДОВА!" має більшу інтенсивность, ніж "їжа тут чудова!"

3. Модифікатори ступеня (також називаються інтенсифікаторами, бустерними словами або прислівниками ступеня) впливають на інтенсивність

настрою шляхом збільшення або зменшення інтенсивності. Наприклад, "Обслуговування тут надзвичайно гарне" має більшу інтенсивність, ніж "Обслуговування тут гарне", тоді як "Обслуговування тут не дуже гарне" знижує інтенсивність.

4. Контрастний зв'язок «але» сигналізує про зміну полярності почуттів, з почуттям тексту, що слідує за кон'юнкцією, яка домінує. "Їжа тут чудово, але обслуговування жахливе" має змішану тональність, а друга половина диктує загальний рейтинг.

5. Вивчаючи триграми, що передує лексичному входженню, що впливає на тональність, правильно оцінюється 90% випадків, коли заперечення змінює полярність тексту. Негативним реченням буде "Їжа тут не така й чудова".

Використовуючи евристику, визначену попередньо, було вибрано 30 базових твітів і створено від шести до десяти варіантів тексту, з однаковою граматичною або синтаксичною функцією, але з різними модифікаторами тональності і проведено невеликомй експеримент з ними. З усіма варіаціями отримано 200 твітів, які було випадково вставлено в новий набір з 800 твітів, аналогічних тим, які використовувалися під час якісного аналізу. Потім 30 незалежних оцінювачів оцінили інтенсивність почуттів усіх 1000 твітів, щоб оцінити вплив цих рис на інтенсивність настроїв.

Таблиця 2.2 показує значення t-критерію Стьюдента, ймовірність, середню різницю у інтенсивонсті між вибірками, і 95-відсотковий довірчий інтервал.

Таблиця 2.2 – Статистичні показники

Тест	t-критерій	Ймовірність	Різниця	95% інтервал
Пунктуація(. або !)	19.02	<2.2e-16	0.291	0.261-0.322
Пунктуація(! або !!)	16.53	2.7e-16	0.215	0.188-0.241
Пунктуація(! або !!)	14.07	1.7e-14	0.208	0.178-0.239
Верхній регістр	28.95	<2.2e-16	0.733	0.682-0.784
Модифікатори ступеня	9.01	6.7e-10	0.293	0.227-0.360

З таблиці 2.2 видно, що для 95% даних, використовуючи знак оклику (відносно точки або взагалі відсутньої пунктуації), інтенсивність збільшилася на 0,261-0,322 з середньою різницею в 0,291 на шкалі оцінки від 1 до 4 (тут ми використовуємо абсолютну шкалу значень для простоти, оскільки не має значення, чи текст є позитивним чи негативним). [13]

2.4.4.3 Порівняння з іншими підходами

Для порівняння з іншими підходами було вибрано наступні набори текстів:

а) записи у соціальній мережі Twitter;

б) відгуки про фільми з сайту rottentomatoes.com;

в) відгуки про техніку з amazon.com;

г) авторські статті з газети New York Times.

В таблиці 2.3 наведено порівняння роботи алгоритму VADER з методами машинного навчання для кожної з виборок. В таблиці приведені значення оцінки F1 для різних вибірок.

Таблиця 2.3 – Порівняння результатів

Алгоритм	Навчальна вибірка	Твіти	Фільми	Техніка	Статті
VADER	-	0.96	0.61	0.63	0.55
Наївний Байес	Твіти	0.84	0.53	0.53	0.42
SVM	Твіти	0.83	0.56	0.55	0.46
Наївний Байес	Фільми	0.56	0.75	0.49	0.44
Наївний Байес	Техніка	0.69	0.55	0.61	0.48
SVM	Техніка	0.64	0.55	0.58	0.42
Наївний Байес	Статті	0.59	0.56	0.51	0.49

З таблиці видно, що алгоритм VADER є кращим у текстах соціальних мереж, однак показує непогані результати і в інших сферах.

Висновки до розділу

В даному розділі розглянуто основні підходи до аналізу тональності тексту. Показана актуальність даної задачі та алгоритми, що її вирішують. Алгоритми можна розділити на два основних класи – методи машинного навчання та методи, що засновані на словниках та правилах. Методи машинного навчання з учителем потребують скалдної та об’ємної роботи для маркування вхідних даних, тому для цієї роботи було вибрано один зі словникових алгоритмів.

РОЗДІЛ 3. МАТЕМАТИЧНІ ОСНОВИ

3.1 Моделі прогнозування часових рядів

Найбільш розповсюдженим методом прогнозування часових рядів є побудова статистичних моделей. Найпростішою з цих моделей є модель AR (Autoregressive, авторегресійна). Модель AR(p) описується формулою 3.1.

$$y(t) = c + \sum_{i=1}^{p} a_i y(t-i) + \varepsilon(t) \tag{3.1}$$

де:

- $y(t)$ – значення часового ряду в момент t;
- c – константа;
- p – порядок моделі;
- a_i – коефіцієнти моделі;
- $\varepsilon(t)$ – білий шум.

Ця модель в якості параметрів для прогнозування використовує попередні значення часового ряду. Ця модель має деякі обмеження на використання, наприклад необхідність того, щоб часовий ряд був стаціонарним.

Логічним продовженням авторегресійної моделі є модель ARMA (Autoregressive Moving Average model, авторегресійна модель з ковзним середнім). Модель ARMA(p, q) описується формулою 3.2.

$$y(t) = c + \sum_{i=1}^{p} a_i y(t-i) + \sum_{j=1}^{q} b_j \varepsilon(t-j) + \varepsilon(t) \tag{3.2}$$

Однак, в даній роботі нас цікавить не авторегресійна складова моделі (тобто залежність курси криптовалют лише від історичних значень), а більше залежність залежної змінної від зовнішнього чиннику (в контексті даної роботи

– настроїв у соціальних мережах). Тому модель ARMA слід доповнити зовнішніми чинниками, які впливають на залежну змінну. Така модель називається ARMAX, описана формулою 3.3

$$y(t) = c + \sum_{i=1}^{p} a_i y(t-i) + \sum_{j=1}^{q} b_j \varepsilon(t-j) + \sum_{k=1}^{n} c_j x(t-k) + \varepsilon(t) \quad (3.3)$$

Цю модель і було вирішено використовувати для системи.

3.2 Алгоритм прогнозування

Для алгоритму прогнозування вхідними параметрами є наступні дані:

- Таблиця з похвилинними даними, яка включає в себе: кількість твітів, середнє значення тональності, курс криптовалюти. Стовпці таблиці: ['sentiment', 'count', 'rate'];
- start_date – дата прогнозу;
- int_length – довжина розбиття даних у хвилинах;
- train_int – розмір навчальної вибірки;
- test_int – кількість інтервалів, на який проводити прогнозування;
- shift – довжина зсуву, тобто, яку кількість попередніх значень тональності та кількості твітів брати до уваги при побудові моделі.

Послідовність дій алгоритму:

- Формування часового ряду з отриманих даних, тобто отримання середнього значення тональності, з розбиттям, рівним int_length хвилин та розміром train_int відносно start_date;

- Обчислення $sentiment_1,\ldots, sentiment_{shift}$ та $count_1,\ldots, count_{shift}$ – створення часових рядів з попередніми значеннями тональності та кількості твітів;
- На мою думку, не є логічним прогнозувати саме курс криптовалют, будемо прогнозувати його зміну за попередній період;
- Нормалізація даних;
- Побудова моделі ARMAX з отриманих після попередніх перетворень часових рядів;
- Прогнозування та аналіз результатів.

Висновки до розділу

В даному розділі розглянуті моделі прогнозування часових рядів, а точніше AR, ARMA та ARMAX. Модель ARMAX застосовується у побудованій системі. Також описаний саме розроблений алгоритм короткострокового прогнозування курсу криптовалют.

РОЗДІЛ 4. АРХІТЕКТУРА ПРОГРАМНОГО ПРОДУКТУ

Для повноцінної роботи системи було вирішено розбити її на два незалежні модулі:

- модуль для збору та аналізу тональності даних соціальних мереж;
- модуль для аналізу курсу криптовалют.

4.1 Модуль для збору даних

Цей модуль виконує наступні задачі:

- збір інформації про курс криптовалют;
- створення потокової системи для отримання даних з соціальної мережі Twitter;
- аналіз тональності інформації, отриманої з вищезгаданого потоку;
- запис отриманої інформації у базу даних.

4.1.1 Вибір мови програмування

Так як збір даних з потоку має бути неперервним, це означає, що модуль має бути оформленим у вигляді веб-додатку. Однією з найбільш розповсюджених мов програмування для веб є Java. Java – об’єктно-орієнтована мова програмування. Ця мова має безліч переваг в порівнянні з іншими – працює майже на всіх операційних системах, має велику кількість бібліотек для роботи з web-ресурсами.

4.1.2 Модулі та бібліотеки

Одним з найбільш простих, стабільних та перевірених способів побудови веб-додатку на Java є Spring Boot. Spring Boot має вбудовані сервлет-контейнери та більшість бібліотек, необхідних для побудови веб-додатку. [14]

Задача організації потокових даних також вирішена у бібліотеці Spring Social. Ця платформа є інтерфейсом доступу до соціальних мереж, а в цій роботі використовувалася її реалізація – Spring Social Twitter.[15]

Для аналізу тональності використовується вищезгаданий алгоритм VADER.[13] Розробники цього алгоритму використовували мову програмування Python для реалізації класифікатора, однак існує також реалізація на мові Java.[16]

Для збереження проаналізованих даних використовується база даних MongoDB[17] та модуль Spring Data MongoDB для підключення до бази даних.[18]

4.1.3 Джерела даних

Для отримання даних з мережі Twitter використовується Twitter Streaming API. Twitter Streaming API дозволяє безкоштовно отримувати потік фільтрованих записів з Twitter. [19] Дані про курс криптовалют беруться з Binance API.[20]

4.1.4 Архітектура модуля

Діаграма класів програмного продукту показана на рис 4.1

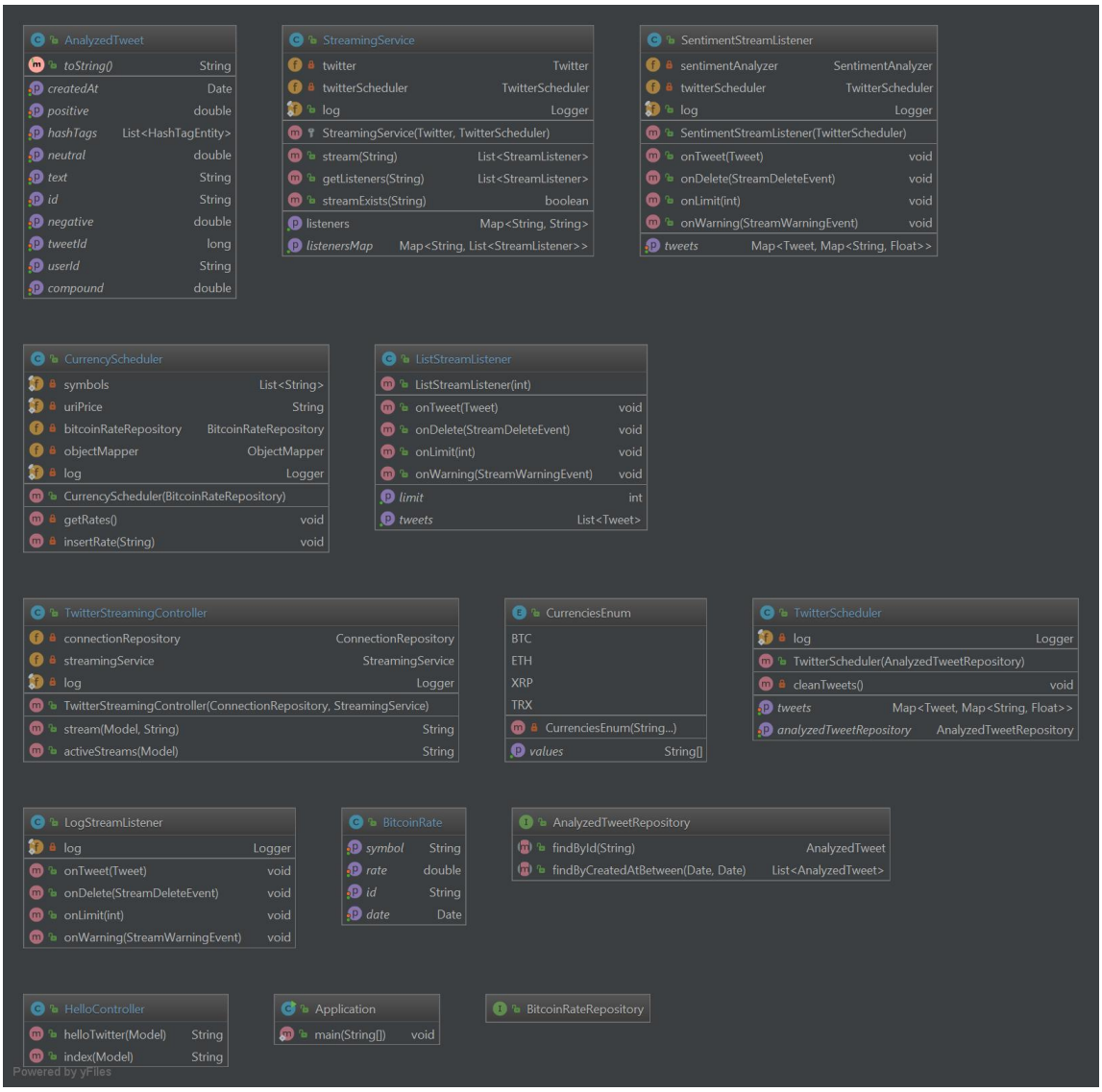


Рисунок 4.1 – Діаграма класів модуля збору даних

4.2 Модуль аналізу даних

Цей модуль виконує наступні задачі:

- зчитування інформації з бази даних;
- побудова статистичних моделей з отриманих даних;
- побудова прогнозу.

4.2.1 Вибір мови програмування

Задачею цього модуля в першу чергу є статистичний аналіз даних, для цієї задачі найкраще підходить мова програмування Python. Ця мова має безліч вбудованих методів для роботи з великими масивами даних, а також достатню кількість різноманітних бібліотек для їх обробки.

4.2.2 Модулі та бібліотеки

Для роботи з даними, що зберігаються у базі даних необхідно мати інтерфейс для підключення до неї. Для цієї задачі використовується PyMongo. PyMongo - це дистрибутив Python, що містить інструменти для роботи з MongoDB, і є рекомендованим способом роботи з MongoDB з Python. [21]

Зберігання і обробка масивів даних реалізована в pandas. pandas – бібліотека з відкритим кодом, BSD-ліцензією, що забезпечує високопродуктивні, прості у використанні структури даних та інструменти аналізу даних для мови програмування Python.[22]

4.3 Результати роботи програми

Для перевірки прогнозу було вибране прогнозування для різної величини інтервалів: 5, 15, 30, 60 хвилин

Для оцінки результатів використовувались наступні метрики:

– MSE – середньоквадратична похибка;

– Процент трендів, які вгадані, тобто, процент випадків коли система прогнозувала зріст курсу, і він дійсно зростав, і навпаки.

Інтервал у 5 хвилин (рис 4.2, синім показана зміна курсу, помаранчевим – спрогнозоване значення):

– MSE – 94.1359654446649;

– Процент спрогнозованих трендів – 0.9.

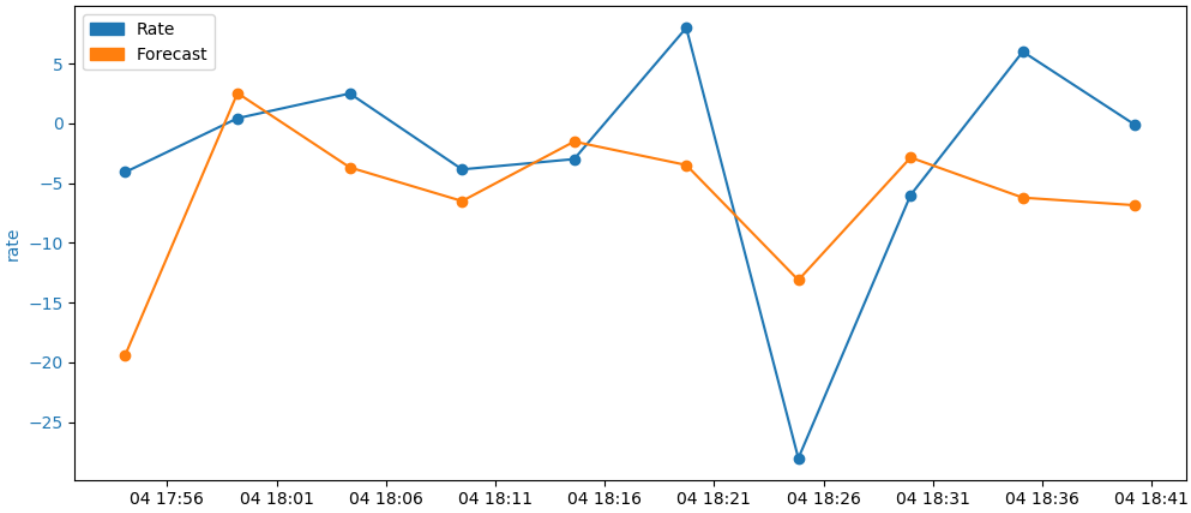


Рисунок 4.2 – Прогноз курсу на 5-хвилинні інтервали

Інтервал у 15 хвилин(рис 4.3, синім показана зміна курсу, помаранчевим – спрогнозоване значення):

– MSE – 608.7813486826228;

– Процент спрогнозованих трендів – 0.6.

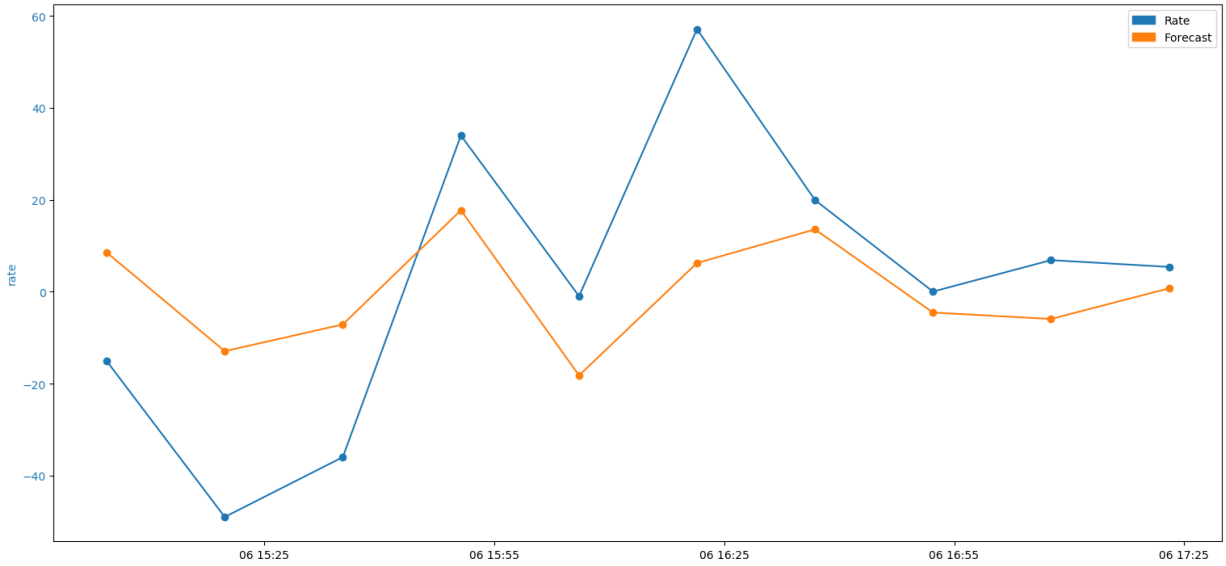


Рисунок 4.3 – Прогноз курсу на 15-хвилинні інтервали

Інтервал у 30 хвилин(рис 4.4, синім показана зміна курсу, помаранчевим – спрогнозоване значення):

– MSE – 903.729038825731;
– Процент спрогнозованих трендів – 0.7.

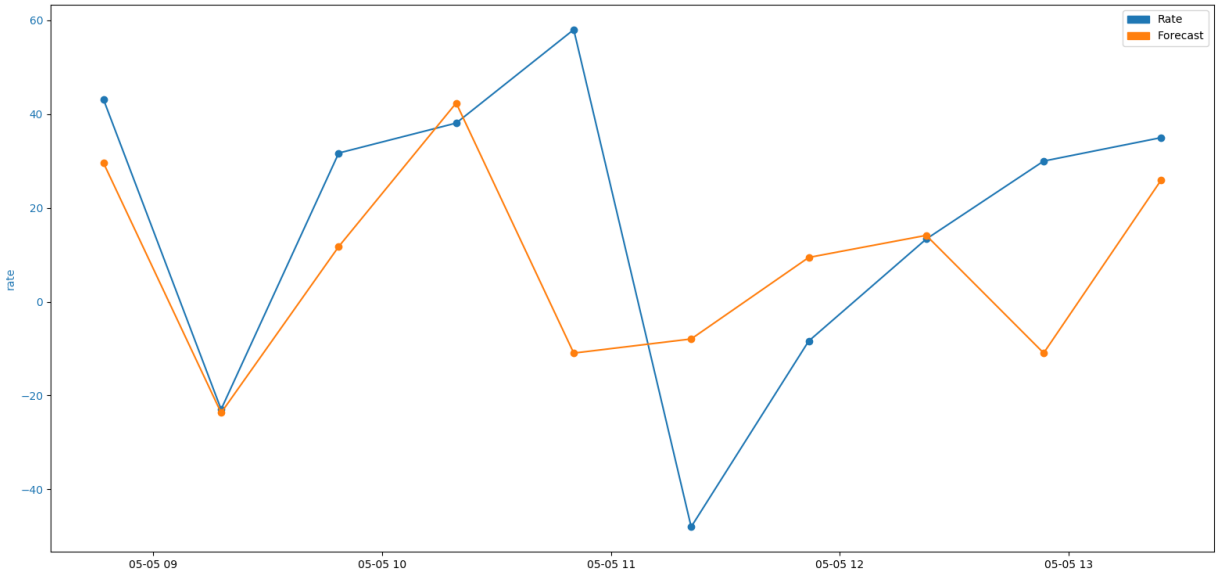


Рисунок 4.4 – Прогноз курсу на 30-хвилинні інтервали

Інтервал у 60 хвилин(рис 4.5, синім показана зміна курсу, помаранчевим – спрогнозоване значення):

– MSE – 3238.5839737925107;

– Процент спрогнозованих трендів – 0.8.

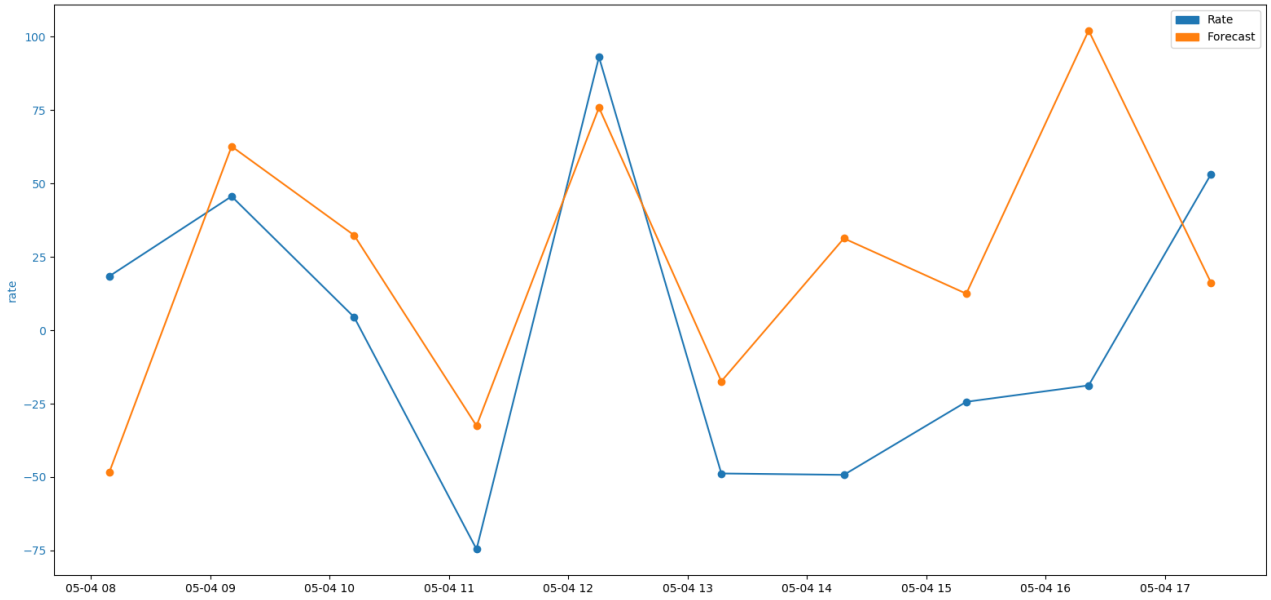


Рисунок 4.5 – Прогноз курсу на 60-хвилинні інтервали

Висновки до розділу

В розділі розглянуто реалізований програмний продукт, що складається з двох окремих модулів: модуля збору даних і аналізу тональності та модуль прогнозування написані, відповідно, на Java та Python. Модулі є абсолютно незалежними, єдиною точкою перетину є база даних, в яку модуль збору даних пише дані, а модуль прогнозування зчитує. Прогноз значень курсу є досить точним, однак його точність може бути покращена за рахунок використання більш складних моделей, наприклад, нейронних мереж. Великим недоліком є те, що система не може розпізнати різкі зміни курсу, які можна побачити на графіках. Втім, на мою думку, такі ситуації неможливо прогнозувати в даній

сфері, адже волатильність курсу криптовалют є занадто високою. Однак, алгоритм досить непогано відслідковує тренди (те, буде курс зростати, чи зменшуватись). До того ж, при великих об’ємах вибірки можна побачити, що точність спадає в залежності від довжини інтервалу. Так, при довжині інтервалу у 5 хвилин середня кількість правильно спрогнозованих трендів складає близько 0.6 (що є досить непоганим в сфері криптовалют). Однак, при збільшенні інтервалу до 60 хвилин середня кількість правильно спрогнозованих трендів знижується до 0.4, тобто прогноз не є цінним. Причина в тому, ща на цьому інтервалі тональність в соціальних мережах не має великого впливу на курс, тобто до системи потрібно додавати інши чинники впливу, такі як кількість і об’єм транзакцій, об’єм ринку та інші.

РОЗДІЛ 5. РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого: головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Далі наведено маркетинговий аналіз стартап проекту. В межах цього етапу:

1) розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
2) аналізуються ринкові можливості щодо його реалізації;
3) на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

5.1 Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

1) зміст ідеї (що пропонується);
2) можливі напрямки застосування;
3) основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
4) чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 5.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 5.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Підписку на розроблену систему прогнозування курсу криптовалют можна продавати користувачам. Також можна використовувати власні кошти та інвестиції для заробляння.	1. Продажа підписки користувачам	Користувачі матимуть змогу знати якісний прогноз курсу криптовалют
	2. Застосування власних коштів та інвестицій	За рахунок власних коштів та інвестицій можна заробляти на коливанні курсу.

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

а) визначення переліку техніко-економічних властивостей та характеристик ідеї;

б) визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

в) проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (табл. 5.2).

Таблиця 5.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторо-на)	N (ней-тральна сторона)	S (сильна сторона)
		Мій проект	FORecast 4u			
1.	Кросплатформе-ність	Можливість використання на різних ОС	Тільки Windows			+
2.	Зручність використання/ орієнтованість на кінцевого споживача	Зручний інтерфейс з підтримкою укр, англ та рос мов.	Тільки англ. мова.			+
3.	Універсальність	Обмежені функції: побудова моделі і прогнозування	Наявні інші функції	+		

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 5.3):

а) за якою технологією буде виготовлено товар згідно ідеї проекту?

б) чи існують такі технології, чи їх потрібно розробити/доробити?

в) чи доступні такі технології авторам проекту?

Таблиця 5.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення програмного забезпечення для побудови моделі та прогнозування	Алгоритми аналізу на нелінійність та нестаціонарність	Наявна	Доступна
		Java	Наявна	Доступна
		Python	Наявна	Доступна
		MongoDB	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: Алгоритми + критеріальна база + Java+Python+MongoDB				

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як Java, Python, MongoDB через їх доступність та безкоштовність.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.4).

Таблиця 5.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	150000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	-
5	Специфічні вимоги до стандартизації та сертифікації	-
6	Середня норма рентабельності в галузі (або по ринку), %	18

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу таблиці 5.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та зформовано орієнтовний перелік вимог до товару для кожної групи (табл. 5.5).

Таблиця 5.5 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Програмне забезпечення для прогнозування курсу криптовалют.	Активні коричтувачі інтернету, люди, які зацікавлені у криптовалютах.	Різний трейдинговий підхід.	Зручний інтерфейс, мала похибка у прогнозуванні, швидкість і надійність у використанні.

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 5.6, 5.7).

Таблиця 5.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок продуктів з кращими характеристиками	Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них саме після виходу на ринок конкурентів. Вдосконалення технічних моментів власного продукту. Обрати нову цільову аудиторію і зосередитися на ній: зниження цін.
2	Невідповідність умовам соціального розвитку	Динамічна зміна соціальних норм чи економічних моментів, що призведе до втрати достовірності прогнозу	Забезпечення гнучкості математичних моделей, адаптація до сучасних умов швидкими темпами
3	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до створюваного ПП

Таблиця 5.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Конкуренція	Відсутність аналогічного продукту для вітчизняного користувача.	Адаптація програмного продукту до вітчизняних особливостей.
2	Поява нових методів прогнозування	З'являться нові методи, що будуть швидше та ефективніше прогнозувати показники	Покращити ПП додаванням нового функціоналу, розширення можливостей
3	Поява нових методів моделювання	З'являться нові методи, що будуть швидше, та більш точно моделювати процеси	Покращити ПП додаванням нового функціоналу, розширення можливостей

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 5.8).

Таблиця 5.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою.	Підтримка якості продукту та постійні нововведення, вдосконалення.
2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу ПП таким чином, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - міжгалузева	Продукт може використовуватись для різних галузей	Постійне вдосконалення продукту, що не має прив'язки до сфери
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами ПП, їх особливостями.	Створити ПП, враховуючи недоліки конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПП, щоб собівартість була нижчою	Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо ці технології відповідають необхідним вимогам якості.

Продовження таблиці 5.8

6. За інтенсивністю - не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 5.9).

Таблиця 5.9 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
	SAS Matlab	Наявність вже існуючих рішень	-	Контроль якості продукту	Наявність більш широкого функціоналу, зручнішого інтерфейсу та авторитет
Висновки:	Досить інтенсивна конкурентна боротьба з вже закріпившимися на ринку гравцями	Є можливості виходу на ринок, але є і конкуренти. Строки – 18 місяців.	-	Клієнти диктують умови роботи на ринку: зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Необхідно випускати ПЗ не гірше, ніж у конкурентів та розширяти функціонал.

За результатами аналізу табл. 5.9 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентноспроможним на ринку.

Цей висновок був врахований при формулюванні переліку факторів конкурентноспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в табл. 5.9, а також із урахуванням характеристик ідеї проекту (табл. 5.2), вимог споживачів до товару (табл. 5.5) та факторів маркетингового середовища (таблиці 5.6, 5.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлено у табл. 5.10

Таблиця 5.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ціна	Більш доступна ціна збільшує кількість потенційних клієнтів
2	Кросплатформність ПП	Можливість використання програмного забезпечення на будь-якій платформі.
3	Орієнтованість на кінцевого споживача	Продукт орієнтований на взаємодію з клієнтом

За визначеними факторами конкурентоспроможності (табл. 5.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 5.11).

Таблиця 5.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з … (назва підприємства)						
			–3	–2	–1	0	+1	+2	+3
1	Ціна	15					*		
2	Кросплатформеність ПП	20			*				
3	Орієнтованість на кінцевого споживача	7					*		

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 5.11). Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками

(прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 5.12 - SWOT-аналіз стартап-проекту

Сильні сторони: Ціна Орієнтованість на кінцевого споживача	Слабкі сторони: Кросплатформеність ПП
Можливості: Конкуренція Поява нових методів прогнозування Поява нових методів моделювання	Загрози: Невідповідність умовам соціального розвитку Зміна потреб користувачів

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. таблицю 5.9, аналіз потенційних конкурентів). Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 5.13).

Таблиця 5.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне розповсюдження створеного ПП	45%	18 місяців
2	Створення ПП з подальшим розповсюдженням за певну оплату	85%	18 місяців
3	Створення вебсайту, на якому можна буде користуватися ПП	75%	16 місяців

Після аналізу було обрано альтернативу №2

5.4 Аналіз ринкової стратегії проекту

За результатами аналізу потенційних груп споживачів було обрано цільові групу, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 5.14).

Таблиця 5.14 - Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1		Визначити потреби кожної з груп, розробити відповідно до них стратегії приваблення клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту (миттєве практичне застосування), орієнтованість на кінцевого споживача	Стратегія диференціації

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 5.15).

Таблиця 5.15 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	«Прешопроходець»	Шукати нових	Ні	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 5.5), а також в залежності від

обраної базової стратегії розвитку (таблиця 5.14) та стратегії конкурентної поведінки (таблиця 5.15) розроблено стратегію позиціонування (таблиця 5.16), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.16 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність та точність результату

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

5.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 5.17 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 5.17 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Швидкість отримання результату	Швидка побудова моделі та створення прогнозу	Відсутність необхідності звертатися до сторонньої особи/компанії для побудови моделі та прогнозу. Дані компанії-користувача, якими оперує ПП, не передаються третім особам, чого вимагає політика безпеки багатьох компаній.
2	Зручність застосування	Не потрібно мати глибоких знань, для того щоб побудувати модель та спрогнозувати показники	ПП сам обирає необхідний та оптимальний метод для побудови моделі та прогнозу. Не потрібно мати глибоких знань у прогнозуванні для того, щоб користуватися ПП
3	Практичність та точність результату	Користувач отримує точні (з малою похибкою розбіжності) результати.	Користувач на виході роботи ПП отримує модель та прогноз, котрі відповідають необхідним показникам достовірності та точності. Отриманий прогноз можна використовувати для створення стратегії розвитку підприємтсва.

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 5.18).

1-й рівень При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 5.18 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Зручність та швидкість отримання практичного результату щодо побудови моделі та прогнозування процесів.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. функція побудови моделі процесу 2. функція побудови прогнозу		
	Якість: достовірність побудови математичної моделі, достовірність побудови прогнозу		
	Пакування: відсутнє		
	Марка: StatLabs «Forec»		
III. Товар із підкріпленням	До продажу: відсутнє		
	Після продажу: персональна підтримка в обслуговуванні за додаткову платню.		
Вихідний код та математична модель будуть закриті. На ідею зареєстровано патент.			

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання за допомогою ноу-хау. Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 5.19). Аналіз проведено експертним методом.

Таблиця 5.19 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1800$	3500$	У всіх трьох груп високий рівень доходів	Базова покупка 1000$ Подальша персональна підтримка в обслуговуванні 150$/місяць

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 5.20)

Таблиця 5.20 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть отримати вигоду та покращити дохідність. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	Встановлення контактів із споживачами і підтримання їх. Формування попиту і стимулювання збуту. Дослідницька робота зі збору маркетингової інформації. Доробка товару, виходячи з потреб конкретного покупця.	Один (від виробника одразу споживачу)	Прямий канал збуту до споживача, мінімізувати збутові витрати розвиток маркетингового спілкування із споживачем

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.21).

Таблиця 5.21 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть отримати вигоду та покращити дохідність. Вони цікавляться сучасними розробками та іноваціями, тому відвідують конференції, інтернет-конференції, семінари.	Конференції, інтернет-конференції, семінари, огляд професійної літератури, інтернет, періодичні видання у різноманітних (профільних) галузях.	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Створення репутації фірмі — виробнику чи посереднику; · збільшення чистого прибутку та рентабельності фірми; · збільшення потоків покупців та обсягів продажу; · стабілізація обсягів продажу в період зменшення попиту.	Шукаєте вірний шлях для розвитку вашої компанії? Досить даремно гаяти час на вгадування вірної стратегії! Користуйтесь «Forec» і світле майбутнє вашій компанії забезпечено!

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу

В даному розділі було проведено аналіз програмного продукту у якості стартап проекту. Можна зазначити, що у проекту є можливість комерціалізації,

оскільки ринок потребує якісний продукт, що надає можливість створювати моделі нелінійних-нестаціонарних процесів.

На ринку наявна монополістична конкуренція, існує декілька фірм-конкурентів, але їх товар дещо відрізняється, тому вихід на ринок не буде легким і потребує грамотної стратегії виходу. Для впровадження ринкової реалізації проекту слід обрати альтернативу, яка передбачає розробку програмного продукту з подальшим розповсюдженням за певну плату.

Можна сказати, що подальший розвиток проекту є доцільним, оскільки він знайде свою цільову аудиторію.

ВИСНОВКИ

В даній роботі розглядається побудова системи прогнозування криптовалют з використанням аналізу тональності новин. Під новинами в контексті даної роботи розуміються записи в соціальній мережі Twitter. Для побудови системи було виконано наступні задачі:

- Досліджено існуючі методи прогнозування курсу криптовалют, виділено їх основні недоліки.
- Досліджено основні методи аналізу тональності тексту, проведено їх порівняння та вибрано метод, який найкраще підходить для вирішення поставленої задачі
- Описано алгоритм короткострокового прогнозування курсу криптовалют, заснований на статистичних моделях
- Реалізовано програмний продукт, що складається з двох модулів:
  - Модуль збору даних та аналізу тональності, написаний на мові Java
  - Модуль прогнозування, написаний на мові Python
- Отримано і проаналізовано результати пронозування

На коротких періодах прогнозування (до 30 хвилин) система правильно прогнозує тренди курсу криптовалют в 60% випадків, однак це число знижується зі зростом інтервалу прогнозування.

Можливий подальший розвиток системи – використання більш складних алгоритмів для прогнозування (наприклад, нейронних мереж), а також розробка алгоритму для довгострокового прогнозування.

ПЕРЕЛІК ПОСИЛАНЬ

1. Vejačka, Martin. Basic Aspects of Cryptocurrencies / Martin Vejačka // Journal of Economy, Business and Financing. - 2014. – no 2. – pp. 75 – 83.
2. Why are cryptocurrencies so volatile? [Електронний ресурс] / Kevin Liao, Computer Science Ph.D. student at UIUC – Режим доступу: https://www.quora.com/Why-are-cryptocurrencies-so-volatile.
3. Giving a twitter bot ability to predict bitcoin value based on historical data [Електронний ресурс] / Ognjen Gatalo – Режим доступу: *https://hackernoon.com/giving-a-twitter-bot-ability-to-predict-bitcoin-value-based-on-historical-data-dbe237c40430*
4. Shah D. Bayesian regression and bitcoin. / Shah D, Zhang K // 52nd Annual Allerton conference on communication, control, and computing. – Allerton. - 2014. – pp. 409–414
5. Bitcoin price prediction algorithm using bayesian regression techniques[Електронний ресурс] - Режим доступу: *https://github.com/panditanvita/BTCpredictor*
6. The future of cryptocurrencies [Електронний ресурс] - Режим доступу: *https://cryptocurrencyhub.io/the-future-of-cryptocurrencies-52e59e4632b*
7. Sentiment analysis for predicting cryptocurrency prices [Електронний ресурс] - Режим доступу: *https://medium.com/@cryptopredicted/sentiment-analysis-for-predicting-cryptocurrency-prices-66282b0ac9a6*
8. What is big data? [Електронний ресурс] - Режим доступу: https://www01.ibm.com/software/data/bigdata/what-is-bigdata.html.
9. Amazing social media statistics [Електронний ресурс] - Режим доступу: *https://socialpilot.co/blog/125- amazing-social-media-statistics-know-2016/*.
10. Hong Kee Sul. Trading on twitter: Using social media sentiment to predict stock returns./ Hong Kee Sul, Alan R Dennis, Lingyao Ivy Yuan // Decision Sciences,

2016. [Електронний ресурс] - Режим доступу: https://onlinelibrary.wiley.com/doi/pdf/10.1111/deci.12229

11. Colianni, Stuart Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis / Stuart Colliani, Stephanie Rosales, Michael Signorotti. [Електронний ресурс] - Режим доступу: http://cs229.stanford.edu/proj2015/029_report.pdf
12. Vimalkumar B. Vaghela Analysis of Various Sentiment Classification Techniques / Vimalkumar B. Vaghela, Bhumika M. Jadav // International Journal of Computer Applications. – 2016. – no 3. – pp. 22-27
13. C.J. Hutto VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text / C.J. Hutto and Eric Gilbert // Eighth International Conference on Weblogs and Social Media (ICWSM-14). – MI. - June 2014.
14. Spring Boot [Електронний ресурс] - Режим доступу: *https://projects.spring.io/spring-boot/*
15. Spring Social Twitter [Електронний ресурс] - Режим доступу: *http://projects.spring.io/spring-social-twitter/*
16. Java port of Python NLTK Vader Sentiment Analyzer. [Електронний ресурс] - Режим доступу: *https://github.com/apanimesh061/VaderSentimentJava*
17. MongoDB [Електронний ресурс] - Режим доступу: *https://www.mongodb.com/*
18. Spring Data MongoDB [Електронний ресурс] - Режим доступу: *https://projects.spring.io/spring-data-mongodb/*
19. Twitter [Електронний ресурс] - Режим доступу: *https://developer.twitter.com/en/docs/tweets/filter-realtime/overview*
20. Binance API [Електронний ресурс] - Режим доступу: *https://github.com/binance-exchange/binance-official-api-docs/blob/master/rest-api.md*
21. PyMongo [Електронний ресурс] - Режим доступу: *https://api.mongodb.com/python/current/*
22. Pandas[Електронний ресурс] - Режим доступу: https://pandas.pydata.org/

ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

HelloController.java

```
package com.ergleb.twitterpredictions.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
@RequestMapping("/")
public class HelloController {

    @RequestMapping(method = RequestMethod.GET)
    public String helloTwitter(Model model) {
        return "redirect:/connect/twitter";
    }

    @RequestMapping(path = "index", method = RequestMethod.GET)
    public String index(Model model) {
        return "connect/twitterConnected";
    }

}
```

TwitterStreamingController.java

```
package com.ergleb.twitterpredictions.controllers;


import com.ergleb.twitterpredictions.services.StreamingService;
import com.ergleb.twitterpredictions.streamlisteners.ListStreamListener;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.social.connect.ConnectionRepository;
import org.springframework.social.twitter.api.StreamListener;
import org.springframework.social.twitter.api.Twitter;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import javax.inject.Inject;
import java.util.List;
import java.util.Map;

@Controller
@RequestMapping("/twitter/stream")
public class TwitterStreamingController {

    private ConnectionRepository connectionRepository;

    private StreamingService streamingService;

    @Inject
```

```
        public TwitterStreamingController(ConnectionRepository
connectionRepository, StreamingService streamingService) {
            this.connectionRepository = connectionRepository;
            this.streamingService = streamingService;
        }

        @RequestMapping("")
        public String stream(Model model, @RequestParam String filter) {
            if  (connectionRepository.findPrimaryConnection(Twitter.class)  ==
null) {
                return "redirect:/connect/twitter";
            }
            log.debug("filter: {}", filter);

            try {
                model.addAttribute("filter", filter);
                List<StreamListener>              listeners              =
streamingService.stream(filter);
                for (StreamListener listener : listeners) {
                    if (listener instanceof ListStreamListener) {
                        model.addAttribute("tweets",     ((ListStreamListener)
listener).getTweets());
                    }
                }
            } catch (Exception ex) {
                log.error("exc: {}", ex);
            }


            return "stream";
        }

        @RequestMapping("/active")
        public String activeStreams(Model model) {
            Map<String,          String>          streamListeners          =
streamingService.getListeners();
            model.addAttribute("streamListeners", streamListeners);
            return "active";
        }

        private        static        final        Logger        log        =
LoggerFactory.getLogger(TwitterStreamingController.class);
    }
```

AnalyzedTweet.java

```
package com.ergleb.twitterpredictions.database.mongo.entity;

import lombok.Getter;
import lombok.Setter;
import lombok.ToString;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;
import org.springframework.social.twitter.api.HashTagEntity;

import java.util.Date;
import java.util.List;
```

```
@Getter
@Setter
@ToString
@Document(collection = "tweets")
public class AnalyzedTweet {

    @Id
    private String id;

    private long tweetId;
    private String text;
    private List<HashTagEntity> hashTags;
    private String userId;
    private Date createdAt;

    private double positive;
    private double negative;
    private double neutral;
    private double compound;
}
```

BitcoinRate.java

```
package com.ergleb.twitterpredictions.database.mongo.entity;


import lombok.Getter;
import lombok.Setter;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

import java.util.Date;

@Document(collection = "rate")
@Getter
@Setter
public class BitcoinRate {

    @Id
    private String id;
    private String symbol;
    private double rate;
    private Date date;
}
```

CurrencyScheduler.java

```
package com.ergleb.twitterpredictions.scheduling;

import com.ergleb.twitterpredictions.database.mongo.entity.BitcoinRate;
import
com.ergleb.twitterpredictions.database.mongo.repository.BitcoinRateRepository;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
```

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

import javax.inject.Inject;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

@Component
public class CurrencyScheduler {

    private static final List<String> symbols = new ArrayList<>();

    static {
        symbols.add("BTCUSDT");
        symbols.add("ETHUSDT");
        symbols.add("ETHBTC");
        symbols.add("XRPBTC");
        symbols.add("TRXBTC");
    }

    private static final String uriPrice =
"https://api.binance.com/api/v3/ticker/price?symbol=";


    private BitcoinRateRepository bitcoinRateRepository;

    private ObjectMapper objectMapper = new ObjectMapper();

    @Inject
    public CurrencyScheduler(BitcoinRateRepository bitcoinRateRepository) {
        this.bitcoinRateRepository = bitcoinRateRepository;
    }

    @Scheduled(fixedDelay = 60000L)
    private void getRates() {
        for (String symbol : symbols) {
            insertRate(symbol);
        }
    }

    private void insertRate(String symbol) {
        RestTemplate restTemplate = new RestTemplate();
        String price = restTemplate.getForObject(uriPrice + symbol,
String.class);

        try {
            JsonNode dataNode = objectMapper.readTree(price);
            double btcPrice = dataNode.get("price").asDouble();
            Date date = new Date();
            BitcoinRate rate = new BitcoinRate();
            rate.setSymbol(symbol);
            rate.setDate(date);
            rate.setRate(btcPrice);
            log.info("price: {}", btcPrice);
            bitcoinRateRepository.insert(rate);
        } catch (IOException ex) {
```

```
                log.error("exception while processing json: {}, ex: {}", price,
ex);
            }
        }

        private        static        final        Logger        log        =
LoggerFactory.getLogger(CurrencyScheduler.class);
    }
```

TwitterScheduler.java

```
    package com.ergleb.twitterpredictions.scheduling;

    import com.ergleb.twitterpredictions.database.mongo.entity.AnalyzedTweet;
    import
com.ergleb.twitterpredictions.database.mongo.repository.AnalyzedTweetRepository;
    import com.vader.sentiment.util.ScoreType;
    import lombok.Getter;
    import lombok.Setter;
    import org.slf4j.Logger;
    import org.slf4j.LoggerFactory;
    import org.springframework.scheduling.annotation.Scheduled;
    import org.springframework.social.twitter.api.Tweet;
    import org.springframework.stereotype.Component;

    import javax.inject.Inject;
    import java.util.HashMap;
    import java.util.List;
    import java.util.Map;
    import java.util.stream.Collectors;


    @Component
    @Getter
    @Setter
    public class TwitterScheduler {
        private        static        final        Logger        log        =
LoggerFactory.getLogger(TwitterScheduler.class);

        @Inject
        public                        TwitterScheduler(AnalyzedTweetRepository
analyzedTweetRepository) {
            this.analyzedTweetRepository = analyzedTweetRepository;
        }

        private AnalyzedTweetRepository analyzedTweetRepository;

        @Getter
        private Map<Tweet, Map<String, Float>> tweets = new HashMap<>();

        @Scheduled(fixedDelay = 5000L)
        private void cleanTweets() {
            log.info("Scheduled op, tweets with polarity: {}", tweets);
            List<AnalyzedTweet>                analyzedTweets                =
tweets.entrySet().stream().map(x -> {
                Tweet tweet = x.getKey();
```

```
                    AnalyzedTweet analyzedTweet = new AnalyzedTweet();
                    analyzedTweet.setTweetId(Long.valueOf(tweet.getId()));
                    analyzedTweet.setText(tweet.getText());
                    analyzedTweet.setHashTags(tweet.getEntities().getHashTags());
                    analyzedTweet.setUserId(tweet.getFromUser());
                    analyzedTweet.setCreatedAt(tweet.getCreatedAt());
                    Map<String, Float> polarity = x.getValue();
                    analyzedTweet.setPositive(polarity.get(ScoreType.POSITIVE));
                    analyzedTweet.setNegative(polarity.get(ScoreType.NEGATIVE));
                    analyzedTweet.setCompound(polarity.get(ScoreType.COMPOUND));
                    analyzedTweet.setNeutral(polarity.get(ScoreType.NEUTRAL));
                    return analyzedTweet;
                }).collect(Collectors.toList());
                log.info("Analyzed tweet list: {}", analyzedTweets);
                analyzedTweetRepository.insert(analyzedTweets);
                tweets = new HashMap<>();
            }
        }
```

StreamingService.java

```
package com.ergleb.twitterpredictions.services;

import com.ergleb.twitterpredictions.scheduling.TwitterScheduler;
import com.ergleb.twitterpredictions.streamlisteners.ListStreamListener;
import
com.ergleb.twitterpredictions.streamlisteners.SentimentStreamListener;
import lombok.Getter;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.social.twitter.api.StreamListener;
import org.springframework.social.twitter.api.Tweet;
import org.springframework.social.twitter.api.Twitter;
import org.springframework.stereotype.Service;

import javax.inject.Inject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Service
public class StreamingService {

    @Getter
    private   Map<String,   List<StreamListener>>   listenersMap   =   new
HashMap<>();

    public List<StreamListener> stream(String filterWords) {

        if (!listenersMap.containsKey(filterWords)) {
            List<StreamListener> streamListeners = new ArrayList<>();
            SentimentStreamListener    sentimentStreamListener    =    new
SentimentStreamListener(twitterScheduler);
            streamListeners.add(sentimentStreamListener);
            streamListeners.add(new ListStreamListener(10));
            log.debug("SentimentSL: {}", sentimentStreamListener);
```

```
                twitter.streamingOperations().filter(filterWords,
streamListeners);
                listenersMap.put(filterWords, streamListeners);
                return streamListeners;
            } else {
                return listenersMap.get(filterWords);
            }
        }

        public Map<String, String> getListeners () {
            Map<String, String> result = new HashMap<>();
            for      (Map.Entry<String,      List<StreamListener>>      entry:
listenersMap.entrySet()) {
                for (StreamListener listener : entry.getValue()) {
                    if (listener instanceof ListStreamListener) {
                        List<Tweet>     tweets     =     ((ListStreamListener)
listener).getTweets();
                        String tweetText = "";
                        if (tweets.size() > 1) {
                            tweetText     =     tweets.get(tweets.size()     -
1).getText();
                        }
                        result.put(entry.getKey(), tweetText);
                    }
                }
            }
            return result;
        }

        public List<StreamListener> getListeners(String filter) {
            return listenersMap.get(filter);
        }

        public boolean streamExists(String filter) {
            return listenersMap.containsKey(filter);
        }

        private Twitter twitter;

        private TwitterScheduler twitterScheduler;

        @Inject
        protected    StreamingService(Twitter    twitter,    TwitterScheduler
twitterScheduler) {
            this.twitter = twitter;
            this.twitterScheduler = twitterScheduler;
        }

        public        static        final        Logger        log        =
LoggerFactory.getLogger(StreamingService.class);

    }
```

ListStreamListener.java

```
package com.ergleb.twitterpredictions.streamlisteners;

import lombok.Getter;
```

```
import org.springframework.social.twitter.api.*;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

@Getter
public class ListStreamListener implements StreamListener {

    private List<Tweet> tweets = new ArrayList<>();
    private final int limit;

    public ListStreamListener(int limit) {
        this.limit = limit;
    }

    @Override
    public void onTweet(Tweet tweet) {
        if (tweet.getLanguageCode().equalsIgnoreCase("en")) {
            tweets.add(tweet);
            if (tweets.size() > limit) {
                tweets.remove(0);
            }

            List<HashTagEntity>             hashTagEntities             =
tweet.getEntities().getHashTags();
            List<String>                    hashTags                    =
hashTagEntities.stream().map(HashTagEntity::getText).collect(Collectors.toList()
);
        }
    }

    @Override
    public void onDelete(StreamDeleteEvent deleteEvent) {

    }

    @Override
    public void onLimit(int numberOfLimitedTweets) {

    }

    @Override
    public void onWarning(StreamWarningEvent warningEvent) {

    }


}
```

SentimentStreamListener.java

```
package com.ergleb.twitterpredictions.streamlisteners;

import com.ergleb.twitterpredictions.scheduling.TwitterScheduler;
import com.vader.sentiment.analyzer.SentimentAnalyzer;
```

```
import com.vader.sentiment.util.ScoreType;
import lombok.Getter;
import lombok.Setter;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.social.twitter.api.StreamDeleteEvent;
import org.springframework.social.twitter.api.StreamListener;
import org.springframework.social.twitter.api.StreamWarningEvent;
import org.springframework.social.twitter.api.Tweet;

import javax.inject.Inject;
import java.util.HashMap;
import java.util.Map;

public class SentimentStreamListener implements StreamListener {

    @Getter
    @Setter
    private Map<Tweet, Map<String, Float>> tweets = new HashMap<>();

    private SentimentAnalyzer sentimentAnalyzer = new SentimentAnalyzer();

    private TwitterScheduler twitterScheduler;

    @Inject
    public SentimentStreamListener(TwitterScheduler twitterScheduler) {
        this.twitterScheduler = twitterScheduler;
    }

    @Override
    public void onTweet(Tweet tweet) {
        try {
            log.trace("onTweet start");
            log.trace("Tweet's text: {}", tweet.getText());
            if (tweet.getLanguageCode().equalsIgnoreCase("en")) {
                sentimentAnalyzer.setInputString(tweet.getText());
                sentimentAnalyzer.setInputStringProperties();
                sentimentAnalyzer.analyze();
                log.debug("tweet:    {},    \n    polarity:    {}",    tweet,
sentimentAnalyzer.getPolarity());
                if
(Math.abs(sentimentAnalyzer.getPolarity().get(ScoreType.COMPOUND)) > 0.2) {
                    twitterScheduler.getTweets().put(tweet,
sentimentAnalyzer.getPolarity());
                }
            }
            log.trace("onTweet end");
        } catch (Exception ex) {
            log.error("error: {}", ex);
        }


    }

    @Override
    public void onDelete(StreamDeleteEvent deleteEvent) {

    }

    @Override
    public void onLimit(int numberOfLimitedTweets) {
```

```
        }

        @Override
        public void onWarning(StreamWarningEvent warningEvent) {
            log.warn("Warning: {}", warningEvent.getMessage());
        }

        public static final Logger log =
LoggerFactory.getLogger(SentimentStreamListener.class);
    }
```

Application.java

```
package com.ergleb.twitterpredictions;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.scheduling.annotation.EnableScheduling;

@SpringBootApplication
@EnableScheduling
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

main.py

```
import configparser
from datetime import datetime, timedelta

import matplotlib.pyplot as plt
import pandas as pd
from pymongo import MongoClient


def get_db(connection_string, db_name):
    client = MongoClient(connection_string)
    db = client[db_name]
    return db


def get_rates(col, symbol):
    rates = col.find({"symbol": symbol})
    return rates


def get_rates_dated(col, symbol, start_date, end_date):
    rates = col.find({"symbol": symbol, "date": {"$lt": end_date, "$gt":
start_date}})
    return rates
```

```
    def get_posts(col):
        return col.count()


    def get_avg_sentiment(col, start_date, end_date):
        agg_string  =  [{"$match":  {"createdAt":  {"$lt":  end_date,  "$gt":
start_date}}},
                        {"$group": {"_id": None, "avg": {"$avg": "$compound"}}}]
        return col.aggregate(agg_string)


    def get_tweets(col, start_date, end_date):
        return col.find({"createdAt": {"$lt": end_date, "$gt": start_date}})


    config = configparser.ConfigParser()
    config.read("mongo_config.ini")
    connection_string = config["mongo"]["connection_string"]
    db = get_db(connection_string, 'test')
    col_rates = db.rate
    # rates = get_rates(col_rates, 'BTCUSDT')
    # for rate in rates:
    #      print(rate)
    start = datetime(2018, 5, 2, 22, 17, 9)
    end = datetime(2018, 5, 5, 11, 30, 14)

    delta = timedelta(minutes=200)
    col_tweets = db.tweets

    # while start < end:
    #     tweets_by_date = get_avg_sentiment(col_tweets, start, start + delta)
    #     print(start + delta)
    #     for tweet in tweets_by_date:
    #         print(tweet)
    #     start = start + delta

    tweets_cursor = get_tweets(col_tweets, start, start + delta)
    rates_cursor = get_rates_dated(col_rates, "BTCUSDT", start, start + delta)
    tweets_df = pd.DataFrame(list(tweets_cursor))
    rates_df = pd.DataFrame(list(rates_cursor))
    tweets_df = tweets_df.sort_values("createdAt")
    print(tweets_df.dtypes)
    rolling = tweets_df["compound"].rolling(window=100).mean()
    rates_df = rates_df.sort_values("date")

    fig, ax1 = plt.subplots()

    color = 'tab:red'
    ax1.set_xlabel('time (s)')
    ax1.set_ylabel('sentiment', color=color)
    ax1.plot_date(tweets_df["createdAt"],        rolling,        color=color,
linestyle="solid")
    ax1.tick_params(axis='y', labelcolor=color)

    ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

    color = 'tab:blue'
    ax2.set_ylabel('rate', color=color)  # we already handled the x-label with
ax1
```

```
      ax2.plot_date(rates_df["date"],       rates_df['rate'],       color=color,
linestyle="solid")
      ax2.tick_params(axis='y', labelcolor=color)

      fig.tight_layout()  # otherwise the right y-label is slightly clipped
      plt.show()
```

Analysis.py

```
      from datetime import datetime, timedelta

      import matplotlib.pyplot as plt
      import pandas as pd
      import statsmodels.api as sm
      import numpy as np
      import sklearn.metrics as metr
      import matplotlib.patches as mpatches



      def data_by_int_length_from_beginning(data, length, int_length):
          ready_data   =   pd.DataFrame(columns=['count',   'sentiment',   'rate',
'date'])
          iteration = 0
          while iteration <= length:
              temp_data   =   data[iteration   *   int_length:(iteration   +   1)   *
int_length]
              if len(temp_data) < int_length:
                  break
              count = 0
              avg = 0.0
              for index, row in temp_data.iterrows():
                  count += row['count']
                  avg += row['sentiment']
              avg /= int_length
              result_dict = {
                  'count': count,
                  'sentiment': avg,
                  'rate': temp_data.iloc[len(temp_data) - 1]['rate'],
                  'date': temp_data.iloc[len(temp_data) - 1]['date']
              }
              ready_data      =      ready_data.append(pd.DataFrame(result_dict,
index=[0]), ignore_index=True)
              iteration += 1

          ready_data['date'] = pd.to_datetime(ready_data['date'])
          ready_data.sort_values("date", inplace=True)
          return ready_data


      def data_by_int_length_from_end(data, length, int_length):
          ready_data   =   pd.DataFrame(columns=['count',   'sentiment',   'rate',
'date'])

          iteration = 0
          while iteration <= length:
              temp_data    =    data[(len(data)    -    (iteration    +    1)    *
int_length):(len(data) - iteration * int_length)]
```

```
            temp_data.index = temp_data.index - max(temp_data.index.values)
            count = 0
            avg = 0.0
            for index, row in temp_data.iterrows():
                count += row['count']
                avg += row['sentiment']
            avg /= int_length
            result_dict = {
                'count': count,
                'sentiment': avg,
                'rate': temp_data.iloc[len(temp_data) - 1]['rate'],
                'date': temp_data.iloc[len(temp_data) - 1]['date']
            }
            ready_data      =      ready_data.append(pd.DataFrame(result_dict,
index=[0]), ignore_index=True)
            iteration += 1

        ready_data['date'] = pd.to_datetime(ready_data['date'])
        ready_data.sort_values("date", inplace=True)
        return ready_data

    def  prepare_data(data,  start_date=datetime(2018,  5,  2,  22,  17,  9),
int_length=1,
                      train_int=20, test_int=5, shift=1):
        data['date'] = pd.to_datetime(data['date'])
        data.sort_values("date", inplace=True)
        #data = data_by_int_length_from_beginning(data, len(data) / int_length,
int_length)
        data = generate_shifts(data, shift=shift)
        test_data = data[(data['date'] > start_date)]
        test_data.index = range(0, len(test_data))
        test_data = test_data[:train_int]
        train_data = data[(data['date'] <= start_date)]
        train_data.index = range(0, len(train_data))
        train_data = data[len(train_data) - test_int:]
        ready_data   =   pd.DataFrame(columns=['count',   'sentiment',   'rate',
'date'])
        #
        #                                test_data                                =
generate_shifts(data_by_int_length_from_beginning(test_data, test_int + shift))
        # train_data = generate_shifts(data_by_int_length_from_end(train_data,
train_int + shift), shift)
        ready_data = train_data.append(test_data)
        return ready_data


    def generate_shifts(data, shift=1):
        for i in range(0, shift):
            data['sentiment' + str(i + 1)] = data['sentiment'].shift(i)
            data['count' + str(i + 1)] = data['count'].shift(i)
        data.dropna(inplace=True)
        data.index = range(0, len(data))
        return data


    def normalize(data, shift=1):
        data['prev_rate'] = data['rate'].shift(-1)
        # data['prev_sentiment'] = data['sentiment'].shift(-shift)
        # print(data)
        data['rate'] = data['rate'].diff()
        max_rate = data['rate'].max()
```

```
        min_rate = data['rate'].min()
        data['rate'] = (data['rate'] - min_rate) / (max_rate - min_rate)
        data['prev_rate'] = data['prev_rate'].diff()
        data['prev_rate']  =  (data['prev_rate']  -  min_rate)  /  (max_rate  -
min_rate)

        # data['sentiment'] = data['sentiment'].diff()
        # data['count'] = data['count'].diff()
        data['count'] = pd.to_numeric(data['count'])
        max_count = data['count'].max()
        min_count = data['count'].min()
        data['count'] = (data['count'] - min_count) / (max_count - min_count)
        for i in range(0, shift):
            data['count' + str(i + 1)] = (data['count' + str(i + 1)] - min_count)
/ (max_count - min_count)
        data['const'] = 1
        data.dropna(inplace=True)
        data.index = data.index - min(data.index)
        return data, min_rate, max_rate


    def generate_sarimax_exog(data, shift=1):
        col_names = []
        for i in range(0, shift):
            col_names.append("sentiment" + str(i + 1))
            col_names.append("count" + str(i + 1))
        print(col_names)
        temp_data = data[col_names]
        temp_data.index = temp_data.index - min(temp_data.index.values)
        return temp_data


    def calc(data, start_date=datetime(2018, 5, 2, 22, 17, 9), int_length=1,
             train_int=20, test_int=5, eps=5, shift=1):
        ready_data      =      prepare_data(data,      start_date=start_date,
int_length=int_length,
                                   test_int=test_int,      train_int=train_int,
shift=shift)
        #ready_data = generate_shifts(ready_data, shift=shift)
        ready_data, min_rate, max_rate = normalize(ready_data, shift=shift)
        data_train = ready_data[:train_int]
        data_test = ready_data[train_int:train_int + test_int]
        model1                    =                   sm.OLS(data_train['rate'],
np.matrix(generate_sarimax_exog(data_train, shift=shift), dtype='float'))
        results1 = model1.fit()
        print(results1.summary())
        data_test = data_test.dropna()
        pred   =   results1.predict(np.matrix(generate_sarimax_exog(data_test,
shift=shift), dtype='float'))
        # print("MSE ", metrics.mean_squared_error(data_test['rate'], pred))

        # print(np.matrix(data_train['rate'].values, dtype='float'))
        #                     print(np.matrix(generate_sarimax_exog(data_train,
shift=shift).values, dtype='float'))

        model2                                                                =
sm.tsa.statespace.SARIMAX(endog=np.array(data_train['rate'].values,
dtype='float'),

exog=np.matrix(generate_sarimax_exog(data_train, shift=shift).values,
                                                          dtype='float'),
```

```
                                                order=(1, 0, 0),
                                                enforce_invertibility=False,
enforce_stationarity=False)
          results2 = model2.fit()
          print(results2.summary())
          forecasts = results2.forecast(test_int,

exog=np.matrix(generate_sarimax_exog(data_test,              shift=shift).values,
dtype='float'))
          # print(forecasts)

          # success_num = 0
          # for i in range(1, test_int):
          #     if abs(pred[train_int + i]) > eps:
          #         if pred[train_int + i] * data_test.iloc[i]['rate'] > 0:
          #             success_num += 1
          #     elif abs(data_test.iloc[i]['rate']) < eps:
          #         success_num += 1
          # success_rate = success_num / test_int
          # print('success_rate', success_rate)

          # color = 'tab:red'
          # ax1.set_xlabel('time (s)')
          # ax1.set_ylabel('sentiment', color=color)
          # ax1.plot_date(data_test["date"], data_test['sentiment'], color=color,
linestyle="solid")
          # ax1.tick_params(axis='y', labelcolor=color)

          # ax2 = ax1.twinx()  # instantiate a second axes that shares the same
x-axis

          unnorm_rate   =   (data_test['rate']   *   (max_rate   -   min_rate)   +
min_rate).values
          print(unnorm_rate)
          unnorm_pred = pred * (max_rate - min_rate) + min_rate
          print(unnorm_pred)
          unnorm_forecasts = forecasts * (max_rate - min_rate) + min_rate
          print(unnorm_forecasts)

          success_num = 0
          for i in range(0, test_int):
              if abs(unnorm_pred[i]) > eps:
                  if unnorm_pred[i] * unnorm_rate[i] > 0:
                      success_num += 1
              elif abs(unnorm_rate[i]) < eps:
                  success_num += 1
          success_rate = success_num / test_int
          print('success_rate without prev', success_rate)

          print("MSE without prev: " + repr(metr.mean_squared_error(unnorm_rate,
unnorm_pred)))
          print("RMSE without prev: " + repr(metr.mean_squared_error(unnorm_rate,
unnorm_pred) ** 0.5))
          print("MAE without prev: " + repr(metr.mean_absolute_error(unnorm_rate,
unnorm_pred)))
          print("R^2   without   prev:   "   +   repr(metr.r2_score(unnorm_rate,
unnorm_pred)))

          success_num = 0
          for i in range(0, test_int):
              if abs(unnorm_forecasts[i]) > eps and abs(unnorm_rate[i]) > eps:
```

```
                if (unnorm_forecasts[i] * unnorm_rate[i]) > 0:
                    success_num += 1
                    print("success")
            elif abs(unnorm_rate[i]) < eps and abs(unnorm_forecasts[i]) < eps:
                success_num += 1
                print("success")
        success_rate = success_num / test_int
        print('success_rate with prev', success_rate)

        print("MSE with prev: " + repr(metr.mean_squared_error(unnorm_rate,
unnorm_forecasts)))
        print("RMSE with prev: " + repr(metr.mean_squared_error(unnorm_rate,
unnorm_forecasts) ** 0.5))
        print("MAE with prev: " + repr(metr.mean_absolute_error(unnorm_rate,
unnorm_forecasts)))
        print("R^2 with prev: " + repr(metr.r2_score(unnorm_rate,
unnorm_forecasts)))

        fig, ax2 = plt.subplots()

        color = 'tab:blue'
        ax2.set_ylabel('rate', color=color)  # we already handled the x-label
with ax1
        ax2.plot_date(data_test["date"], unnorm_rate, color=color,
linestyle="solid")
        #       ax2.plot_date(data_test["date"],       result['pred'].values,
color='tab:green', linestyle="solid")
        ax2.plot_date(data_test["date"], unnorm_forecasts, color='tab:orange',
linestyle="solid")
        ax2.tick_params(axis='y', labelcolor=color)
        rate_patch = mpatches.Patch(color='tab:blue', label='Rate')
        fcst_patch = mpatches.Patch(color='tab:orange', label='Forecast')

        plt.legend(handles=[rate_patch, fcst_patch])
        plt.show()


        result = pd.DataFrame(columns=['date', 'rate','pred','fcst'])
        result['date'] = data_test['date']
        result['rate'] = unnorm_rate
        result['pred'] = unnorm_pred
        result['fcst'] = unnorm_forecasts

        print(result)
        return result

    # start = datetime(2018, 5, 3, 1, 22, 9)
    # df = pd.read_csv("data.csv")
    # calc(df, start_date=start, train_int=20, int_length=5, eps=5)
    # start = datetime(2018, 5, 4, 12, 22, 9)
    # df = pd.read_csv("data.csv")
    # calc(df, start_date=start, train_int=20, int_length=15, eps=5)
    # start = datetime(2018, 5, 3, 2, 22, 9)
    # df = pd.read_csv("data.csv")
    # calc(df, start_date=start, train_int=20, int_length=30, eps=5)
    start = datetime(2018, 5, 3, 20, 30, 9)
    int_length = 60
    test_int = 10
    eps = 10

    df = pd.read_csv("data.csv")
```

```
      df = data_by_int_length_from_beginning(df, 999999, int_length)
      result = pd.DataFrame(columns=['date', 'rate', 'pred', 'fcst'])
      for i in range(0, 30):
          result = result.append(calc(df, start_date=start, train_int=20,
int_length=int_length, eps=eps, shift=4, test_int=test_int))
          start = start + timedelta(minutes=int_length * test_int)
      result.index = range(0, len(result))
      print(result)

      fig, ax2 = plt.subplots()

      color = 'tab:blue'
      ax2.set_ylabel('rate', color=color)  # we already handled the x-label with
ax1
      ax2.plot_date(result["date"],         result['rate'],         color=color,
linestyle="solid")
      #          ax2.plot_date(data_test["date"],          result['pred'].values,
color='tab:green', linestyle="solid")
      ax2.plot_date(result["date"],      result['fcst'],      color='tab:orange',
linestyle="solid")
      ax2.tick_params(axis='y', labelcolor=color)
      rate_patch = mpatches.Patch(color='tab:blue', label='Rate')
      fcst_patch = mpatches.Patch(color='tab:orange', label='Forecast')
      plt.legend(handles=[rate_patch, fcst_patch])
      success_num = 0
      for i in range(0, len(result)):
          if abs(result['pred'].values[i]) > eps:
              if result['pred'].values[i] * result['rate'].values[i] > 0:
                  success_num += 1
          elif abs(result['rate'].values[i]) < eps:
              success_num += 1
      success_rate = success_num / len(result)
      print('success_rate without prev', success_rate)

      print("MSE             without             prev:             "             +
repr(metr.mean_squared_error(result['rate'].values, result['pred'].values)))
      print("RMSE             without             prev:             "             +
repr(metr.mean_squared_error(result['rate'].values,   result['pred'].values)   **
0.5))
      print("MAE             without             prev:             "             +
repr(metr.mean_absolute_error(result['rate'].values, result['pred'].values)))
      print("R^2  without  prev:  "  +  repr(metr.r2_score(result['rate'].values,
result['pred'].values)))

      success_num = 0
      for i in range(0, len(result)):
          if abs(result['fcst'].values[i]) > eps:
              if (result['fcst'].values[i] * result['rate'].values[i]) > 0:
                  success_num += 1
                  print("success")
          elif abs(result['rate'].values[i]) < eps:
              success_num += 1
              print("success")
      success_rate = success_num / len(result)
      print('success_rate with prev', success_rate)

      print("MSE             with             prev:             "             +
repr(metr.mean_squared_error(result['rate'].values, result['fcst'].values)))
      print("RMSE             with             prev:             "             +
repr(metr.mean_squared_error(result['rate'].values,   result['fcst'].values)   **
0.5))
```

```
      print("MAE              with              prev:              "              +
repr(metr.mean_absolute_error(result['rate'].values, result['fcst'].values)))
      print("R^2   with   prev:   "   +   repr(metr.r2_score(result['rate'].values,
result['fcst'].values)))

      plt.show()
```

ДОДАТОК Б. СЛАЙДИ ПРЕЗЕНТАЦІЇ

Система прогнозування курсу криптовалют на основі аналізу тональності новин

Єрохін Г. В. КА-65м

Актуальність роботи

- Об’єм ринку криптовалют є дуже широким. Капіталізація Bitcoin станом на 17.04.2018 становить $134,137,594,838, а кількість криптовалют з загальною капіталізацією >$1,000,000,000 – 24. Цей показник продовжує рости, так само, як і кількість людей, які вкладають свої кошти у криптовалюти. Саме тому задача прогнозування курсу є актуальною.

Існуючі підходи до розв’язку задачі

- В даний момент запропоновані декілька підходів до задачі прогнозування курсу криптовалют:
  - Пошук К найближчих сусідів
  - Використання Байесівської регресії (MIT, 2014)
- Однак, у цих алгоритмів є проблема – вони беруть до уваги лише минулі значення курсу, але не ситуацію в даний момент часу. А найкращим джерелом інформації в сфері криптовалют є новини та соціальні мережі.

Постановка задачі

- Задача складається у наступному:
  - Зібрати історичні дані з соціальних мереж та курсу криптовалют
  - Провести аналіз тональності записів
  - Побудувати модель прогнозування курсу криптовалюти на основі отриманих тональностей

Мета, предмет та об'єкт

- Мета роботи – створити систему прогнозу курсу криптовалют, її реалізувати та дослідити.
- Об’єкт дослідження – коливання курсу криптовалют
- Предмет дослідження – методи виявлення тональності тексту, методи прогнозування курсів валют.

Математичні основи

- Для аналізу тональності текстів використовується алгоритм VADER. Він використовує словники, що створені людьми, а також словник емотиконів (смайликів) та акронімів(абревіатур). Потім ці лексичні особливості об’єднуються з урахуванням п'яти загальних правил, які втілюють граматичні та синтаксичні конвенції для вираження та підкреслення інтенсивності настроїв.
- Алгоритм повертає оцінку тексту у вигляді масиву з чотирьох полей: позитивні, негативні та нейтральні слова, а також збірна оцінка тексту.

Математичні основи

- Прогнозування курсу криптовалют будується на основі моделі ARMAX:

$$y(t) = c + \sum_{i=1}^{p} a_i y(t-i) + \sum_{j=1}^{q} b_j \varepsilon(t-j) + \sum_{k=1}^{n} c_j x(t-k) + \varepsilon(t)$$

Математичні основи

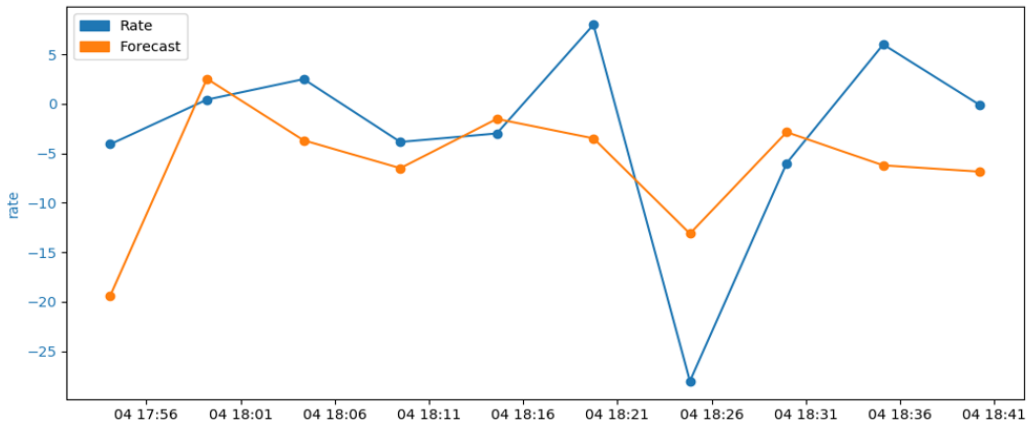
- Робота розробленої системи полягає в наступному
  - Формування часового ряду з отриманих даних, тобто отримання середнього значення тональності, з розбиттям, рівним int_length хвилин та розміром train_int відносно start_date;
  - Обчислення $sentiment_1$,…,$sentiment_{shift}$ та $count_1$,…,$count_{shift}$ – створення часових рядів з попередніми значеннями тональності та кількості твітів;
  - На мою думку, не є логічним прогнозувати саме курс криптовалют, будемо прогнозувати його зміну за попередній період;
  - Нормалізація даних;
  - Побудова моделі ARMAX з отриманих після попередніх перетворень часових рядів;
  - Прогнозування та аналіз результатів.

Система

- Побудована система складається з двох модулів:
  - Модуль для збору даних та оцінки тональності тексту, написаний на мові Java
  - Модуль для прогнозування курсу криптовалют за отриманим значенням тональності, написаний на мові Python
- Модулі є абсолютно незалежними, єдиною точкою перетину є база даних MongoDB, в яку перший модуль записує дані, а других збирає

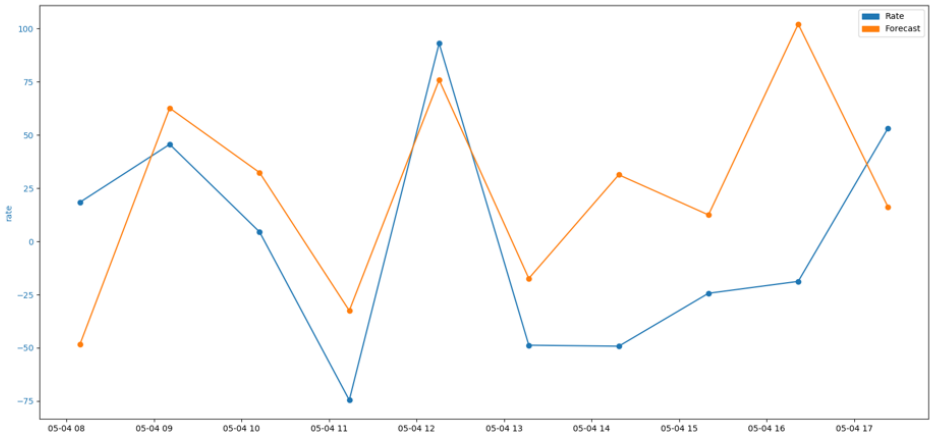
Отримані результати

- Прогноз на 5-хвилинні інтервали:



Отримані результати

- Прогноз на 60-хвилинні інтервали:



Отримані результати

- При довжині інтервалу у 5 хвилин середня кількість правильно спрогнозованих трендів складає близько 0.6 (що є досить непоганим в сфері криптовалют). Однак, при збільшенні інтервалу до 60 хвилин середня кількість правильно спрогнозованих трендів знижується до 0.4, тобто прогноз не є цінним. Причина в тому, ща на цьому інтервалі тональність в соціальних мережах не має великого впливу на курс, тобто до системи потрібно додавати інши чинники впливу, такі як кількість і об’єм транзакцій, об’єм ринку та інші.

Наукова новизна

- Розроблено спосіб прогнозу курсу криптовалют, у якому використовується аналіз тональності новин як зовнішній чинник, що впливає на курс.

Подальший розвиток

- Масштабування системи
- Довгострокові прогнози
- Продаж підписки на сервіс
- Пошук інвесторів та використання системи з власними коштами